**Reporting Tips and Tricks**

Reporting is an essential part of the penetration testing process but, if poorly managed, can become very tedious and prone to mistakes. One key aspect of reporting is that we should be working on building our report from the onset. This starts with our organizational structure/notetaking setup, but there are times when we may be running a long discovery scan where we could fill out templated parts of the report such as contact information, client name, scope, etc. While testing, we can be writing up our Attack Chain and each finding with all of the required evidence so we don't have to scramble to recapture evidence after the assessment is over. Working as we go will ensure that our report isn't rushed and comes back from QA with loads of changes in red.

**Templates**

This should go without saying, but we shouldn't be recreating the wheel with every report we write. It's best to have a blank report template for every assessment type we perform (even the more obscure ones!). If we are not using a reporting tool and just working in old-fashioned MS Word, we can always build a report template with macros and placeholders to fill in some of the data points we fill out for every assessment. We should work with blank templates every time and not just modify a report from a previous client, as we could risk leaving another client's name in the report or other data that does not match our current environment. This type of error makes us look amateur and is easily avoidable.

**MS Word Tips & Tricks**

Microsoft Word can be a pain to work with, but there are several ways we can make it work for us to make our lives easier, and in our experience, it's easily the least of the available evils. Here are a few tips & tricks that we've gathered over the years on the road to becoming an MS Word guru. First, a few comments:

* The tips and tricks here are described for Microsoft Word. Some of the same functionality may also exist in LibreOffice, but you'll have to [preferred search engine] your way around to figure out if it's possible.
* Do yourself a favor, use Word for Windows, and explicitly avoid using Word for Mac. If you want to use a Mac as your testing platform, get a Windows VM in which you can do your reporting. Mac Word lacks some basic features that Windows Word has, there is no VB Editor (in case you need to use macros), and it cannot natively generate PDFs that look and work correctly (it trims the margins and breaks all of the hyperlinks in the table of contents), to name a few.
* There are many more advanced features like font-kerning that you can use to crank your fancy to 11 if you'd like, but we're going to try to stay focused on the things that improve efficiency and will leave it to the reader (or their marketing department) to determine specific cosmetic preferences.

Let's cover the basics:

* Font styles
  + You should be getting as close as you possibly can to a document without any "direct formatting" in it. What I mean by direct formatting is highlighting text and clicking the button to make it bold, italics, underlined, colored, highlighted, etc. "But I thought you "just" said we're only going to focus on stuff that improves efficiency." We are. If you use font styles and you find that you've overlooked a setting in one of your headings that messes up the placement or how it looks, if you update the style itself, it updates "all" instances of that style used in the entire document instead of you having to go manually update all 45 times you used your random heading (and even then, you might miss some).
* Table styles
  + Take everything I just said about font styles and apply it to tables. Same concept here. It makes global changes much easier and promotes consistency throughout the report. It also generally makes everyone using the document less miserable, both as an author and as QA.
* Captions
  + Use the built-in caption capability (right-click an image or highlighted table and select "Insert Caption...") if you're putting captions on things. Using this functionality will cause the captions to renumber themselves if you have to add or remove something from the report, which is a GIGANTIC headache. This typically has a built-in font style that allows you to control how the captions look.
* Page numbers
  + Page numbers make it much easier to refer to specific areas of the document when collaborating with the client to answer questions or clarify the report's content (e.g., "What does the second paragraph on page 12 mean?"). It's the same for clients working internally with their teams to address the findings.
* Table of Contents
  + A Table of Contents is a standard component of a professional report. The default ToC is probably fine, but if you want something custom, like hiding page numbers or changing the tab leader, you can select a custom ToC and tinker with the settings.
* List of Figures/Tables
  + It's debatable whether a List of Figures or Tables should be in the report. This is the same concept as a Table of Contents, but it only lists the figures or tables in the report. These trigger off the captions, so if you're not using captions on one or the other, or both, this won't work.
* Bookmarks
  + Bookmarks are most commonly used to designate places in the document that you can create hyperlinks to (like an appendix with a custom heading). If you plan on using macros to combine templates, you can also use bookmarks to designate entire sections that can be automatically removed from the report.
* Custom Dictionary
  + You can think of a custom dictionary as an extension of Word's built-in AutoCorrect feature. If you find yourself misspelling the same words every time you write a report or want to prevent embarrassing typos like writing "pubic" instead of "public," you can add these words to a custom dictionary, and Word will automatically replace them for you. Unfortunately, this feature does not follow the template around, so people will have to configure their own.
* Language Settings
  + The primary thing you want to use custom language settings for is most likely to apply it to the font style you created for your code/terminal/text-based evidence (you did create one, right?). You can select the option to ignore spelling and grammar checking within the language settings for this (or any) font style. This is helpful because after you build a report with a bunch of figures in it and you want to run the spell checker tool, you don't have to click ignore a billion times to skip all the stuff in your figures.
* Custom Bullet/Numbering
  + You can set up custom numbering to automatically number things like your findings, appendices, and anything else that might benefit from automatic numbering.
* Quick Access Toolbar Setup
  + There are many options and functions you can add to your Quick Access Toolbar that you should peruse at your leisure to determine how useful they will be for your workflow, but we'll list a few handy ones here. Select File > Options > Quick Access Toolbar to get to the config.
  + Back - It's always good to click on hyperlinks you create to ensure they send you to the right place in the document. The annoying part is getting back to where you were when you clicked so you can keep working. This button takes care of that.
  + Undo/Redo - This is only useful if you don't use the keyboard shortcuts instead.
  + Save - Again, useful if you don't use the keyboard shortcut instead.
  + Beyond this, you can set the "Choose commands from:" dropdown to "Commands Not in the Ribbon" to browse the functions that are more difficult to perform.
* Useful Hotkeys
  + F4 will apply the last action you took again. For example, if you highlight some text and apply a font style to it, you can highlight something else to which you want to apply the same font style and just hit F4, which will do the same thing.
  + If you're using a ToC and lists of figures and tables, you can hit Ctrl+A to select all and F9 to update all of them simultaneously. This will also update any other "fields" in the document and sometimes does not work as planned, so use it at your own risk.
  + A more commonly known one is Ctrl+S to save. I just mention it here because you should be doing it often in case Word crashes, so you don't lose data.
  + If you need to look at two different areas of the report simultaneously and don't want to scroll back and forth, you can use Ctrl+Alt+S to split the window into two panes.
  + This may seem like a silly one, but if you accidentally hit your keyboard and you have no idea where your cursor is (or where you just inserted some rogue character or accidentally typed something unprofessional into your report instead of Discord), you can hit Shift+F5 to move the cursor to where the last revision was made.
  + There are many more listed [here](https://support.microsoft.com/en-us/office/keyboard-shortcuts-in-word-95ef89dd-7142-4b50-afb2-f762f663ceb2), but these are the ones that I've found have been the most useful that aren't also obvious.

**Automation**

When developing report templates, you may get to a point where you have a reasonably mature document but not enough time or budget to acquire an automated reporting platform. A lot of automation can be gained through macros in MS Word documents. You will need to save your templates as .dotm files, and you will need to be in a Windows environment to get the most out of this (Mac Word's VB Editor may as well not exist). Some of the most common things you can do with macros are:

* Create a macro that will throw a pop-up for you to enter key pieces of information that will then get automatically inserted into the report template where designated placeholder variables are:
  + Client name
  + Dates
  + Scope details
  + Type of testing
  + Environment or application names
* You can combine different report templates into a single document and have a macro go through and remove entire sections (that you designate via bookmarks) that don't belong in a particular assessment type.
  + This eases the task of maintaining your templates since you only have to maintain one instead of many
* You may also be able to automate quality assurance tasks by correcting errors made often. Given that writing Word macros is basically a programming language on its own (and could be a course all by itself), we leave it to the reader to use online resources to learn how to accomplish these tasks.

**Reporting Tools/Findings Database**

Once you do several assessments, you'll start to notice that many of the environments you target are afflicted by the same problems. If you do not have a database of findings, you'll waste a tremendous amount of time rewriting the same content repeatedly, and you risk introducing inconsistencies in your recommendations and how thoroughly or clearly you describe the finding itself. If you multiply these issues by an entire team, the quality of your reports will vary wildly from one consultant to the next. At a minimum, you should maintain a dedicated document with sanitized versions of your findings that you can copy/paste into your reports. As discussed previously, we should constantly strive to customize findings to a client environment whenever it makes sense but having templated findings saves a ton of time.

However, it is time well spent to investigate and configure one of the available platforms designed for this purpose. Some are free, and some must be paid for, but they will most likely pay for themselves quickly in the amount of time and headache you save if you can afford the initial investment.

| **Free** | **Paid** |
| --- | --- |
| [Ghostwriter](https://github.com/GhostManager/Ghostwriter) | [AttackForge](https://attackforge.com/) |
| [Dradis](https://dradisframework.com/ce/) | [PlexTrac](https://plextrac.com/) |
| [Security Risk Advisors VECTR](https://github.com/SecurityRiskAdvisors/VECTR) | [Rootshell Prism](https://www.rootshellsecurity.net/why-prism/) |
| [WriteHat](https://github.com/blacklanternsecurity/writehat) |  |

**Misc Tips/Tricks**

Though we've covered some of these in other module sections, here is a list of tips and tricks that you should keep close by:

* Aim to tell a story with your report. Why does it matter that you could perform Kerberoasting and crack a hash? What was the impact of default creds on X application?
* Write as you go. Don't leave reporting until the end. Your report does not need to be perfect as you test but documenting as much as you can as clearly as you can during testing will help you be as comprehensive as possible and not miss things or cut corners while rushing on the last day of the testing window.
* Stay organized. Keep things in chronological order, so working with your notes is easier. Make your notes clear and easy to navigate, so they provide value and don't cause you extra work.
* Show as much evidence as possible while not being overly verbose. Show enough screenshots/command output to clearly demonstrate and reproduce issues but do not add loads of extra screenshots or unnecessary command output that will clutter up the report.
* Clearly show what is being presented in screenshots. Use a tool such as [Greenshot](https://getgreenshot.org/) to add arrows/colored boxes to screenshots and add explanations under the screenshot if needed. A screenshot is useless if your audience has to guess what you're trying to show with it.
* Redact sensitive data wherever possible. This includes cleartext passwords, password hashes, other secrets, and any data that could be deemed sensitive to our clients. Reports may be sent around a company and even to third parties, so we want to ensure we've done our due diligence not to include any data in the report that could be misused. A tool such as Greenshot can be used to obfuscate parts of a screenshot (using solid shapes and not blurring!).
* Redact tool output wherever possible to remove elements that non-hackers may construe as unprofessional (i.e., (Pwn3d!) from CrackMapExec output). In CME's case, you can change that value in your config file to print something else to the screen, so you don't have to change it in your report every time. Other tools may have similar customization.
* Check your Hashcat output to ensure that none of the candidate passwords is anything crude. Many wordlists will have words that can be considered crude/offensive, and if any of these are present in the Hashcat output, change them to something innocuous. You may be thinking, "they said never to alter command output." The two examples above are some of the few times it is OK. Generally, if we are modifying something that can be construed as offensive or unprofessional but not changing the overall representation of the finding evidence, then we are OK, but take this on a case-by-case basis and raise issues like this to a manager or team lead if in doubt.
* Check grammar, spelling, and formatting, ensure font and font sizes are consistent and spell out acronyms the first time you use them in a report.
* Make sure screenshots are clear and do not capture extra parts of the screen that bloat their size. If your report is difficult to interpret due to poor formatting or the grammar and spelling are a mess, it will detract from the technical results of the assessment. Consider a tool such as Grammarly or LanguageTool (but be aware these tools may ship some of your data to the cloud to "learn"), which is much more powerful than Microsoft Word's built-in spelling and grammar check.
* Use raw command output where possible, but when you need to screenshot a console, make sure it's not transparent and showing your background/other tools (this looks terrible). The console should be solid black with a reasonable theme (black background, white or green text, not some crazy multi-colored theme that will give the reader a headache). Your client may print the report, so you may want to consider a light background with dark text, so you don't demolish their printer cartridge.
* Keep your hostname and username professional. Don't show screenshots with a prompt like azzkicker@clientsmasher.
* Establish a QA process. Your report should go through at least one, but preferably two rounds of QA (two reviewers besides yourself). We should never review our own work (wherever possible) and want to put together the best possible deliverable, so pay attention to the QA process. At a minimum, if you're independent, you should sleep on it for a night and review it again. Stepping away from the report for a while can sometimes help you see things you overlook after staring at it for a long time.
* Establish a style guide and stick to it, so everyone on your team follows a similar format and reports look consistent across all assessments.
* Use autosave with your notetaking tool and MS Word. You don't want to lose hours of work because a program crashes. Also, backup your notes and other data as you go, and don't store everything on a single VM. VMs can fail, so you should move evidence to a secondary location as you go. This is a task that can and should be automated.
* Script and automate wherever possible. This will ensure your work is consistent across all assessments you perform, and you don't waste time on tasks repeated on every assessment.

**Client Communication**

Strong written and verbal communication skills are paramount for anyone in a penetration testing role. During our engagements (from scoping until final report delivery and review), we must remain in constant contact with our clients and serve appropriately in our role as trusted advisors. They are hiring our company and paying a lot of money for us to identify issues in their networks, give remediation advice, and also to educate their staff on the issues we find through our report deliverable. At the start of every engagement, we should send a start notification email including information such as:

* Tester name
* Description of the type/scope of the engagement
* Source IP address for testing (public IP for an external attack host or the internal IP of our attack host if we are performing an Internal Penetration Test)
* Dates anticipate for testing
* Primary and secondary contact information (email and phone)

At the end of each day, we should send a stop notification to signal the end of testing. This can be a good time to give a high-level summary of findings (especially if the report will have 20+ high-risk findings) so the report does not entirely blindside the client. We can also reiterate expectations for report delivery at this time. We should, of course, be working on the report as we go and not leave it 100% to the last minute, but it can take a few days to write up the entire attack chain, executive summary, findings, recommendations, and perform self-QA checks. After this, the report should go through at least one round of internal QA (and the people responsible for QA probably have lots of other things to do), which can take some time.

The start and stop notifications also give the client a window for when your scans and testing activities were taking place in case they need to run down any alerts.

Aside from these formal communications, it is good to keep an open dialogue with our clients and build and strengthen the trusted advisor relationship. Did you discover an additional external subnet or subdomain? Check with the client to see if they'd like to add it to the scope (within reason and provided it does not exceed the time allotted for testing). Did you discover a high-risk SQL injection or remote code execution flaw on an external website? Stop testing and formally notify the client and see how they would like to proceed. A host seems down from scanning? It happens, and it's best to be upfront about it than try to hide it. Got Domain Admin/Enterprise Admin? Give the client a heads up in case they see alerts and get nervous or so they can prepare their management for the pending report. Also, at this point, let them know that you will keep testing and looking for other paths but ask them if there is anything else they'd like you to focus on or servers/databases that should still be limited even with DA privileges that you can target.

We should also discuss the importance of detailed notes and scanner logging/tool output. If your client asks if you hit a specific host on X day, you should be able to, without a doubt, provide documented evidence of your exact activities. It stinks to get blamed for an outage, but it's even worse if you get blamed for one and have zero concrete evidence to prove that it was not a result of your testing.

Keeping these communication tips in mind will go a long way towards building goodwill with your client and winning repeat business and even referrals. People want to work with others who treat them well and work diligently and professionally, so this is your time to shine. With excellent technical skills and communication skills, you will be unstoppable!

**Presenting Your Report - The Final Product**

Once the report is ready, it needs to go through review before delivery. Once delivered, it is customary to provide the client with a report review meeting to either go over the entire report, the findings alone, or answer questions that they may have.

**QA Process**

A sloppy report will call into question everything about our assessment. If our report is a disorganized mess, is it even possible that we performed a thorough assessment? Were we careless and left a trail of destruction in our wake that the client will have to spend time they don't have to clean up? Let's ensure our report deliverable is a testament to our hard-earned knowledge and hard work on the assessment and adequately reflects both. The client isn't going to see most of what you did during the assessment.

The report is your highlight reel and is honestly what the client is paying for!

You could have executed the most complex awesome attack chain in the history of attack chains, but if you can't get it on paper in a way that someone else can understand, it may as well have never happened at all.

If possible, every report should undergo at least one round of QA by someone who isn't the author. Some teams may also opt to break up the QA process into multiple steps (e.g., QA for technical accuracy and then QA for proper styling and cosmetics). It will be up to you, your team, or your organization to choose the right approach that works for the size of your team. If you are just starting on your own and don't have the luxury of having someone else review your report, I would strongly recommend walking away from it for a while or sleeping on it and reviewing it again at a minimum. Once you read through a document 45 times, you start overlooking things. This mini-reset can help you catch things you didn't see after you had been staring at it for days.

It is good practice to include a QA checklist as part of your report template (remove it once the report is final). This should consist of all the checks the author should make regarding content and formatting and anything else that you may have in your style guide. This list will likely grow over time as you and your team's processes are refined, and you learn which mistakes people are most prone to making. Make sure that you check grammar, spelling, and formatting! A tool such as Grammarly or LanguageTool is excellent for this (but make sure you have approval). Don't send a sloppy report to QA because it may get kicked back to you to fix before the reviewer even looks at it, and it can be a costly waste of time for you and others.

A quick note about online grammar correction tools: As a means to "learn" more and improve the accuracy of the tool, these will often send pieces of whatever data it's reading back "home", which means if you're writing a report with confidential client vulnerability data in it, you might be breaching some sort of MSA or something unwittingly. Before using tools like this, it's important to look into their functionality and whether this sort of behavior can be disabled.

If you have access to someone that can perform QA and you begin trying to implement a process, you may soon find that as the team grows and the number of reports being output increases, things can get difficult to track. At a basic level, a Google Sheet or some equivalent could be used to help make sure things don't get lost, but if you have many more people (like consultants AND PMs) and you have access to a tool like Jira, that could be a much more scalable solution. You'll likely need a central place to store your reports so that other people can get to them to perform the QA process. There are many out there that would work, but choosing the best one is outside the scope of this course.

Ideally, the person performing QA should NOT be responsible for making significant modifications to the report. If there are minor typos, phrasing, or formatting issues to address that can be done more quickly than sending the report back to the author to change, that's likely fine. For missing or poorly illustrated evidence, missing findings, unusable executive summary content, etc., the author should bear the responsibility for getting that document into presentable condition.

You obviously want to be diligent about reviewing the changes made to your report (turn Track Changes on!) so that you can stop making the same mistakes in subsequent reports. It's absolutely a learning opportunity, so don't squander it. If it's something that happens across multiple people, you may want to consider adding that item to your QA checklist to remind people to address those issues before sending reports to QA. There aren't many better feelings in this career than when the day comes that a report you wrote gets through QA without any changes.

It may be considered strictly a formality, but it's reasonably common to initially issue a "Draft" copy of the report to the client once the QA process has been completed. Once the client has the draft report, they should be expected to review it and let you know whether they would like an opportunity to walk through the report with you to discuss modifications and ask questions. If any changes or updates need to be made to the report after this conversation, they can be made to the report and a "Final" version issued. The final report is often going to be identical to the draft report (if the client does not have any changes that need to be made), but it will just say "Final" instead of "Draft ."It may seem frivolous, but some auditors will only consider accepting a final report as an artifact, so it could be quite important to some clients.

**Report Review Meeting**

Once the report has been delivered, it’s fairly customary to give the client a week or so to review the report, gather their thoughts, and offer to have a call to review it with them to collect any feedback they have on your work. Usually, this call covers the technical finding details one by one and allows the client to ask questions about what you found and how you found it. These calls can be immensely helpful in improving your ability to present this type of data, so pay careful attention to the conversation. If you find yourself answering the same questions every time, that could indicate that you need to tweak your workflow or the information you provide to help answer those questions before the client asks them.

Once the report has been reviewed and accepted by both sides, it is customary to change the DRAFT designation to FINAL and deliver the final copy to the client. From here, we should archive all of our testing data per our company's retention policies until a retest of remediated findings is performed at the very least.

**Wrap Up**

These are just some tips and tricks we have collected over the years. Many of these are common sense. This [post](https://blackhillsinfosec.com/how-to-not-suck-at-reporting-or-how-to-write-great-pentesting-reports/) by the awesome team at Black Hills Information Security is worth a read. The goal here is to present the most professional deliverable possible while telling a clear story based on our hard work during a technical assessment. Put your best foot forward and create a deliverable you can be proud of. You spent many hours relentlessly pursuing Domain Admin. Apply that same zeal to your reporting, and you'll be a rockstar. In the last sections of this module, we'll discuss opportunities for practicing our documentation and reporting skills.