Analyzing and Reporting Pen Test Results

- Analyze Pen Test Data
- Develop Recommendations for Mitigation Strategies
- Write and Handle Reports
- Conduct Post-Report-Delivery Activities

Pen Test Data Collection (Slide 1 of 2)

- Ensure sensitive data you gathered during testing doesn't fall into the wrong hands.
  - Addresses
  - Network maps
  - Security details
  - Vulnerabilities

Pen Test Data Collection (Slide 2 of 2)

- Record of all activities you performed will help you identify activities performed as part of testing and those of an actual attacker.
  - Access to secure areas.
  - Web app compromise.
  - Social engineering attacks.
  - Compromising the network with various attacks.
  - Pivoting deeper into the network.
  - Stealing files.
  - Defacing internal sites.
  - Evading detection.

Pen Test Data Categorization

- You categorized assets earlier to determine an approach to exploitation.
- Now you should categorize the pen test results.
- Categorize data in a way that makes sense to both you and the client.
- Categorize data in terms of the type of asset they relate to.
  - Example: Successful SQL injection is a software issue.
  - Consider subcategories, like web app issues as a subcategory of software issues.
- Also categorize based on severity level of vulnerabilities and weaknesses.
  - High-priority items impact the most people, systems, and data.
  - Low-priority items impact a few people, systems, or data.

Prioritization of Results (Slide 1 of 2)

- Work with the client to prioritize the results of testing.
- Most common approach is to use terms to label severity.
  - Critical, high, medium, low, etc.
  - Can also use a number scale, e.g., 1 to 10.
- What seems to be most urgent might not be as urgent based on client needs.
  - Compliance needs.
  - Legacy and/or specialized hardware needs.
  - Example: PCI DSS compliance might be paramount despite more severe vulnerabilities.

Prioritization of Results (Slide 2 of 2)

- Factors to consider might include:
  - PII
  - PHI
  - Network accessibility
  - Building accessibility
  - Data accessibility
- Many factors can influence prioritization.
- More nuance than just labeling something based on its CVSS score.

Guidelines for Analyzing Pen Test Data

- Gather all of the data you have collected.
  - Identify all of the activities you performed to help determine which attacks were carried out by you and which are from attackers.
- Ensure proper handling of all data so it doesn't fall into the wrong hands.
- Categorize data based on the needs of the client.
  - This is most often based on the severity level, but could be based on other factors if the client needs it to be.
- Prioritize the results.
  - Work with the client to identify which items need to be dealt with first.

Suggested Solutions Regarding People

- Implement technical controls.
- Have management set the security tone and lead by example.
- Train people in proper security measures.
- Constant reinforcement and reminders.
- Implement penalties for non-compliance.
- Reward groups that have no incidents.
- Avoid complacency.
- Give users a sense of ownership in the process.

Suggested Solutions Regarding Processes

- Implement technical controls.
- Have managers take an active role.
- Review processes.
- Put KPIs in place.
- Update processes when needed.

Suggested Solutions Regarding Technology (Slide 1 of 2)

- Implementing mitigation solutions using technology often involves a direct cost that the organization needs to budget for.
  - Management tries to get the maximum value out of investments.
  - They might be reluctant to spend more money on technology solutions.
- Mitigation strategies and techniques include:
  - Have IT run monthly vulnerability scans.
  - Have annual security audits and pen tests.
  - Have KPIs that management can use at a glance to see security effectiveness of new technology.
  - Follow the 80/20 rule in risk reduction.
    - 80% of vulnerabilities can be remediated with 20% of the cost and effort.
    - Implement multiple layers of security, each targeting at least 80% of coverage.

Suggested Solutions Regarding Technology (Slide 2 of 2)

- Technology solutions to consider include:
  - Counter downgrade attacks by allowing servers to only use the latest version of TLS and not permitting legacy SSL.
  - To counter SSL strip, configure servers to use HSTS.
  - To counter ARP poisoning, write static ARP tables or implement an IDS.

People, Processes, and Technology

- Balance technology with processes and people.
- Consider ease of use against the need for security.
  - If security procedure is too complicated, users will find ways to bypass it.
- Password cracking is often due to people, process, and technology problems in concert.
  - Password policy is in writing only, not implemented in technology.
  - Passwords that are too simple are easily cracked.
  - Passwords that are too complicated are often written down.

Categories of Findings

- Shared local administrator credentials.
- Weak password complexity.
- Plaintext passwords.
- No multi-factor authentication.
- SQL injection, XSS, other code injection.
- Unnecessary open services.
- Physical intrusion.

End-User Training

- Cybersecurity training for all employees.
- Users should be able to identify why it is important for everyone to do their part in keeping the organization and its assets secure.
- Training should include:
  - How to spot threats they might encounter on the job.
  - The consequences of succumbing to threats.
  - Tools to mitigate threats.
- IT department should be made aware of any suspicious devices.

Password Hashing and Encryption

- Don't allow developers to hard-code credentials in apps.
- Hash stored passwords rather than storing them in plaintext.
- Use strong hash functions, like SHA-256 and bcrypt.
- Avoid weak hash functions, like MD5 and SHA-1.
- Use network access protocols that encrypt passwords in transit.
  - SSH instead of Telnet, HTTPS instead of HTTP, FTPS instead of FTP, etc.
- Ensure network access protocols are using strong ciphers, like AES-256 and RC6.
- Avoid network access protocols that use weak ciphers, like DES and 3DES.
- Disallow services that allow their cryptographic protocols to be negotiated down.
- Ensure security solutions can monitor/manage unencrypted traffic in the network.

Multi-Factor Authentication (Slide 1 of 2)

- Relatively affordable for organizations.
- Feasible for even smaller businesses to adopt.
- Useful in circumstances where user authenticates to a certain system.
  - Systems that provide critical access to company resources.
  - Systems that provide access to PII.
  - Systems that provide access to personal activities; e.g., online banking.
- Even with password policies, users still choose weak passwords.
  - Easy to guess or easy to crack with a wordlist.

Multi-Factor Authentication (Slide 2 of 2)

- MFA compensates for password weaknesses.
- Example: Limited-time security code sent to user's phone over SMS.
  - "Something you have" factor.
  - Many people own smartphones or are issued them by their employer.
- Other methods used in MFA:
  - Smart cards.
  - Hardware tokens/keyfobs.
  - Biometric fingerprint scanners.

Input Sanitization (Slide 1 of 3)

The process of stripping user-supplied input of unwanted or untrusted data so that the application can safely process that input.

- Most common approach to mitigating code injection.
  - Particularly XSS and SQL injection.
- Any form that echoes input or stores it should sanitize the data first.
- Several tactics can be considered sanitization.
  - Each has its own purpose and applies to different attacks.
- For XSS, escaping HTML is most effective.
  - Prevents special characters from being processed by browser.
  - Also called encoding.
  - Substitutes special characters with entities.
  - Example: < becomes `<` when encoded.
  - Encoding command depends on language used.

Input Sanitization (Slide 2 of 3)

- PHP:

```
<?php
function my_func($input) {
    echo htmlspecialchars($input, ENT_QUOTES, 'UTF-8');
}
?>
<!DOCTYPE html>
<html>
    <body>
        <?php my_func('<script>alert("XSS attack successful!");</script>'; ?>
    </body>
</html>
```

Input Sanitization (Slide 3 of 3)

- Encodes input so any instances of special characters are turned into entities.
  - Characters: `& " ' < >`
  - Malicious string turns into: `<script>alert(&quot;XSS attack successful!&quot;);</script>`
  - Browser won't run script.
- Works in most cases, but not all.
  - Doesn't work if app needs to accept HTML input.
  - Use language-based sanitization libraries.

Parameterized Queries (Slide 1 of 2)

The technique of processing SQL input by incorporating placeholders for some of a query's parameters.

- Web app binds actual values to parameters in different statement.
- Example: Quotation mark would be interpreted literally.
  - Not interpreted as if part of query structure.
  - `x' OR 'x'='x` in user name field forces database to match this user name literally.
- Most effective means of preventing SQL injection.
- How you implement will depend on language used.
- Example: PHP uses PDO as abstraction layer for processing database content.

Parameterized Queries (Slide 2 of 2)

```
<?php
$prod_name = ""
$prod_desc = ""

// Code to connect to database
...

// Prepare statement
$stmt = $db_conn->prepare("INSERT INTO products (prod_name, prod_desc) VALUES (:prod_name,
:prod_desc)");
$stmt->bindParam(':prod_name', $prod_name);
$stmt->bindParam(':prod_desc', $prod_desc);

?>
```

- INSERT INTO query is prepared, creating a template for database to parse.
  - Parsed template stored without executing.
- Input values are bound to each parameter and transmitted after query.
- Input values plugged into template, preventing SQL injection.

System Hardening

- Harden hardware and software before adding it to the network.
- Assume the hardware or software is unsafe.
  - Research and identify any known issues.
  - Perform testing to identify any additional vulnerabilities.
- Hardening techniques:
  - Consulting industry standards organizations for best practices/guidelines.
    - ISO, SANS, NIST, CIS, etc.
  - Installing patches/updates that are available.
  - Incorporating patch/change management processes.
  - Ensuring systems have firewalls/anti-malware solutions.
  - Ensuring firewalls are configured to uphold least privilege.
  - Disabling unneeded ports and services.
  - Uninstalling unneeded software.
  - Segmenting hosts on the network.

Mobile Device Management (Slide 1 of 2)

The process of tracking, controlling, and securing the organization's mobile infrastructure.

- Usually web-based centralized platform.
- Can enforce policies, manage apps, data, etc., on all mobile devices in network.
- Worthwhile investment for organizations whose mobile infrastructure is at risk.

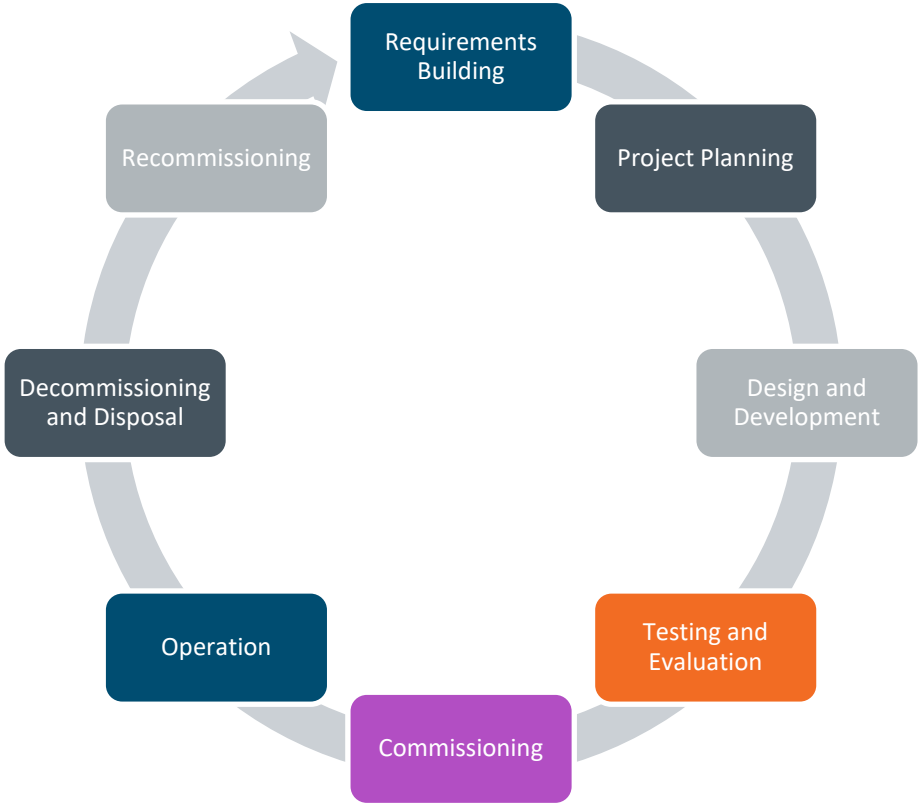
Mobile Device Management (Slide 2 of 2)

- Common features:
  - Pushing out updates to devices.
  - Enrolling/authenticating devices.
  - Enforcing a security policy layer on devices.
  - Locating devices.
  - Configuring devices with access-controlled profiles.
  - Sending out push notifications to devices.
  - Enabling devices to use remote access.
  - Enabling remote lock/wipe.
  - Constructing encrypted containers on devices in which to keep sensitive data.

Secure Software Development (Slide 1 of 3)

- Security should not be an afterthought for software.
  - Whether developed in-house or by a third party.
- Security should be an active component in the development process.
  - Not something applied reactively when issues arise.
- Follow an SDLC.
  - Focuses on design, development, and maintenance of software.
  - Passes through several phases.
  - Security should be incorporated at each phase.
- Example: Testing phase should include fuzzing techniques.
  - Identifies faults in app.
  - Done before pushing app into production.
- SDLC ensures there are no gaps in software security at any point.
  - From beginning phases all the way to the end.

Secure Software Development (Slide 2 of 3)



Secure Software Development (Slide 3 of 3)

- Follow best coding practices.
- Write code that:
  - Is clear and easy for other developers to grasp.
  - Has useful and informative documentation.
  - Is easy to incorporate in the build process.
  - Is highly extensible.
  - Has as few external dependencies as possible.
  - Is concise.
  - Relies on well-established techniques.
  - Integrates well with test harnesses.
  - Closely aligns with design requirements.
- Actively avoid insecure coding practices mentioned previously.

Guidelines for Developing Recommendations for Mitigation Strategies (Slide 1 of 2)

- Consider mitigation in terms of people, processes, and technology.
- Recommend strategies for common findings:
  - Shared local administrator credentials: Randomize credentials or use LAPS.
  - Weak passwords: Configure minimum password requirements and use password filters.
  - Plaintext passwords: Use protocols that hash/encrypt passwords.
  - No multi-factor authentication: Implement/require multi-factor authentication.
  - XSS attacks: Encode/escape special HTML characters.
  - SQL injection: Parameterize queries.
  - Unnecessary open services: Perform system hardening.
  - Physical intrusion: Incorporate guards, cameras, motion alarms, etc.

Guidelines for Developing Recommendations for Mitigation Strategies (Slide 2 of 2)

- Recommend end-user training to mitigate social engineering.
- Recommend system hardening techniques to secure hosts.
- Recommend MDM solutions for mobile infrastructure security.
- Recommend SDLC and best coding practices for secure software development.

Activity: Recommending Mitigation Strategies

- Your report must include more than findings.
- You need to offer suggestions to GCPG.
  - They had you test their systems so the problems could be fixed, not just identified.
- You'll give them recommendations on how to mitigate risks.
- You'll include mitigations in your report.

Data Normalization (Slide 1 of 2)

The process of reducing redundancy and increasing integrity to create a cohesive set of data.

- A term usually associated with databases.
- Can also be applied to pen test reports.
  - Can be stored in a database, in text documents, etc., as agreed upon with the client.
  - Reduce redundancy and increase integrity no matter the format.
- Example: Consolidate data from disparate sources that describes the same event.
  - Reduces confusion.

Data Normalization (Slide 2 of 2)

- Reports are likely to be used by a variety of audiences.
  - They all need to be able to understand your findings and recommendations.
  - Board members might only get the executive summary and some high-level data.
  - End users might only get information that pertains to their job.
  - Administrators might get highly technical information.
- Normalize data in reports to make it clear to the target audience.
- Minimize noise in report data.

Report Structure

- Executive summary
- Methodology
- Findings and remediation
- Metrics and measures
- Risk rating
- Conclusion
- Supporting evidence

		Impact		
		Low	Moderate	High
Likelihood	High	Low	Moderate	High
	Moderate	Low	Moderate	Moderate
	Low	Low	Low	Low

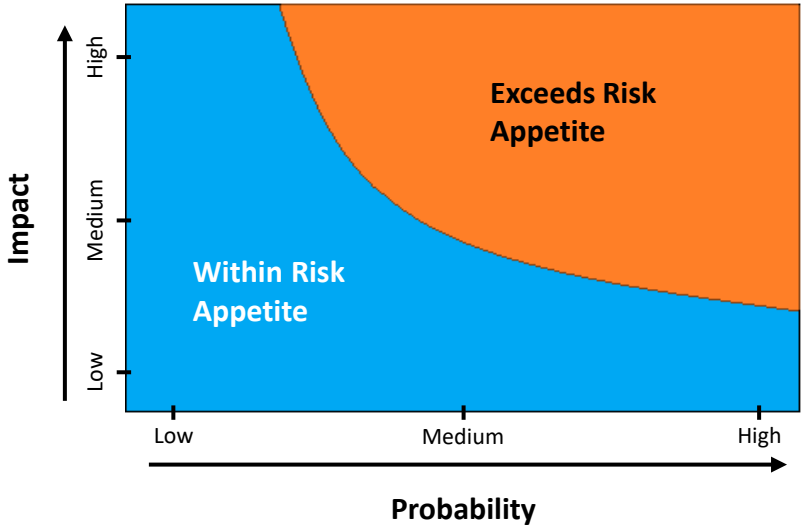
Risk Appetite (Slide 1 of 2)

The amount and type of potential vulnerabilities and threats the organization is willing to tolerate and endure.

- Organization needs to balance:
  - How much risk it is willing to endure.
  - How much it would cost to mitigate the risk.
  - How difficult it would be to implement mitigation strategies.
- Key stakeholders determine risk appetite by answering questions:
  - What losses would be catastrophic to the organization?
  - What assets can be unavailable and still enable the client to function, and for how long?
  - What must be available at all times, and cannot be made publicly accessible?
  - Might circumstances result in personal harm to any stakeholders?

Risk Appetite (Slide 2 of 2)

- Pen test report should account for risk appetite.
  - Determine level of risk of a vulnerability (probability x impact).
  - Compare results to client's appetite.
  - Will help client better understand the impact of risks.



Report Storage (Slide 1 of 2)

- You and client must not allow the report to fall into the wrong hands.
  - Contains highly detailed information about vulnerabilities.
  - Contains other sensitive data.
- Store reports on a secure server.
- Don't pass the report via external drives.
- Implement access control on file system.
  - Only appropriate personnel should be able to view the report.

Report Storage (Slide 2 of 2)

- Some parts need to be available to different personnel.
  - Store pieces and parts in a repository.
  - Apply granular access control to repository.
- Encrypt report file in storage.
- Determine storage time for report.
- Implement document control measures:
  - A cover page.
  - Document properties.
  - Version control information.

Report Handling

- Maintain the confidentiality of reports and their contents.
- Maintain the integrity of reports and their contents.
- Ensure reports are always available to the relevant audience.
- Ensure reports are secure in transit across a network.
- Minimize the transmission of reports across a public network like the Internet.
- Ensure reports are secure in storage.
- Protect reports and their contents from accidental disclosure.
- Maintain audit logs for users accessing reports.
- Maintain a chain of custody when transferring ownership of reports.
- Maintain version control for changes to reports.

Report Disposition

The formal process of transferring the report to the care or possession of the primary authorized recipient.

- You are giving the report to the client.
- At this point, the client becomes responsible for the report and its contents.
- It should include:
  - All documentation.
  - Multiple copies (electronic and printed).
  - Acknowledgements and sign-off by the recipient.
- The client's authorized recipient distributes copies as needed.
  - If requests are made to the pen test team, refer them to the authorized recipient.
- You should move (not copy) your version of the report to backup.
- Remove the report from your active work area when done.
  - Prevents data leakage if work area is compromised.

Guidelines for Writing and Handling Reports (Slide 1 of 2)

- Normalize data to reduce redundancy and increase integrity.
- Consider including the following sections in your report:
  - Executive summary
  - Methodology
  - Findings and remediation
  - Metrics and measures
  - Risk rating
  - Conclusion
  - Supporting evidence

Guidelines for Writing and Handling Reports (Slide 2 of 2)

- Work with the client to determine their risk appetite.
- Write your report to speak to the client's risk appetite.
- Determine the file format for the report.
- Determine where the report will be securely stored
- Follow best practices for securely handling the report.
- Determine how to hand the report off to the client.

Post-Engagement Cleanup Tasks (Slide 1 of 2)

- In cases where exploits destabilize live systems, clean up directly after.
- Otherwise, you can clean up at the end.
- Cleanup prevents leftover artifacts from being used in real exploitation.

Post-Engagement Cleanup Tasks (Slide 2 of 2)

- Common cleanup tasks:
  - Delete new files you created.
  - Remove credentials/accounts you created.
  - Restore original configurations you modified.
  - Restore original files you modified.
  - Restore log files you deleted.
  - Restore original log files you modified.
  - Remove backdoors.
  - Remove additional tools.
  - Purge sensitive data exposed in plaintext.
  - Restore a clean backup copy of compromised apps.
- Consider automating cleanup through scripts.
  - You'll need to have kept meticulous records of your exploits.

Removal of Credentials

- Removing credentials, shells, and tools is not always a simple task.
  - Exploits might be deeply embedded in target systems.
  - Breadth of exploits might be difficult to track/manage.
- Not all authentication systems are alike.
  - You can log on locally and delete local credentials easily.
  - Removing AD credentials is not so easy.
- Removing an AD account from a workstation won't remove it from domain.
  - You need to access DC to delete account.
  - Attacker could leverage domain account to sign in to a domain computer.
- Some credentials are tightly integrated into systems.
  - Systems that require strong audit trails/change history.
  - Not easy to delete these accounts.
  - System must preserve change integrity.
  - May need to remove account directly from database.

Removal of Shells and Other Tools

- Shells are probably hidden on target systems in multiple ways.
- Remove startup values in **HKLM**/**HKCU Run** Registry keys.
- Remove scheduled tasks in Task Scheduler/`crontab` file.
  - Shell might not be currently running, but lying dormant.
- Remove Netcat/other backdoor binaries.
- Remove other tools that enable compromise.
  - Payloads, keyloggers, scanner agents, etc.
  - Some loaded in memory are automatically removed on reboot.
  - Some require manual uninstallation.
  - Shred whenever possible.
- Avoid collateral damage.
  - Only remove test accounts, not legitimate user accounts.
  - Don't remove tools/software that are crucial to system operations.
  - Leave target systems in the state you found them.

Client Acceptance (Slide 1 of 2)

- After finishing the test and writing the report, plan to discuss with the client.
- You'll need to get confirmation from the client that:
  - The test is complete.
  - The report's findings are accurate.
- Use the meeting to discuss details with the client.
  - Anything that needs to be clarified/changed in the report.
- Gaining client's acceptance is of paramount importance.
  - They won't be satisfied just because they read your report.
- Client must be convinced the test was worthwhile from a business standpoint.
  - Test must have truly met objectives set out in planning phase.
  - Example: Provide CBA of implementing recommended mitigations.

Client Acceptance (Slide 2 of 2)

- Client may wish to assess how closely test adhered to established scope.
- Client may benefit from a better understanding of your testing methodologies.
- Client may voice concerns about how test was handled.
- You must work with client to address their concerns.
- You must prove the test was conducted in their best interests.

Attestation of Findings (Slide 1 of 2)

Attestation: The process of providing evidence that the findings detailed in the pen test report are true.

- You are attesting that you believe the info in the report is authentic.
- The most significant component of gaining client acceptance.
  - Client must believe what you say about their business is accurate.
- Many organizations won't just trust your word.
  - Even if you've gained a good reputation.
  - You must be prepared to prove your claims.

Attestation of Findings (Slide 2 of 2)

- Proof can come in many forms.
  - Depends on what's being proven.
  - Example: Exfiltrating sensitive data on a server as proof of compromise.
  - Example: Giving client a live demonstration of connecting to a reverse shell backdoor.
  - Example: Showing client packet capture containing plaintext data in transit.
- Threshold of evidence differs between organizations.
  - Some might be content with screenshots of compromise.
- You must communicate with your client to identify their needs.

Lessons Learned (Slide 1 of 2)

- You should identify lessons learned during the project.
- Debrief with the pen test team.
  - You'll likely uncover what did/did not work well in the test.
- What you learn will influence future tests.
- Primary goal of an LLR/AAR is to improve your processes/tools.
- Failing to learn your lesson can lead to:
  - Repeating the same mistakes.
  - Inefficient use of time.
  - Inaccurate findings/conclusions.
  - Harder time of gaining client acceptance.

Lessons Learned (Slide 2 of 2)

- Ask and answer the following questions:
  - What about the test went well?
  - What about the test didn't go well, or didn't go as well as planned?
  - What can the team do to improve itself for future client engagements?
  - What new vulnerabilities, exploits, etc., did the team learn about?
  - Do you need to change your approach or testing methodology?
  - How will you remediate any issues that you identified?

Follow-Up Actions

- Scheduling additional tests with the client organization.
- Working with the security team that will implement your recommended mitigations.
- Checking back with the client to see how their mitigation efforts are going.
- Researching/testing new vulnerabilities that your team discovered during the test.
- Researching vulnerabilities that the team couldn't recommend a mitigation tactic for.
- Informing the organization if a mitigation tactic is eventually found.

Guidelines for Conducting Post-Report-Delivery Activities

- Verify that you have removed any remaining artifacts of the test.
- Be prepared to have a discussion about the contents of the report.
- Gain the client's acceptance of the test and your reported findings/conclusions.
- Find out what the client needs as far as proof.
- Provide proof of your tests as needed.
- Draft a lessons learned report by asking yourself what did or did not go well.
- Identify areas of improvement for the pen test team's processes and tools.
- Identify any follow-up actions that need to be performed.
- Identify who will be performing these actions.

Reflective Questions

1. What are common lessons learned in pen tests that you've taken part in? What do you expect will work, and what won't?

2. What sorts of proof of compromise would you expect to have to supply?