Chad Ballay

CYBR410-342N

10/05/2020

**Hands On Project 10-1**

**Create a table that includes a rotating schedule for the 12 months of security testing**

I was neck deep in researching and creating my own but I found a crowd sourced answer that covers much of what I was thinking of and many things I hadn’t thought of. This would be the likely better answer than one created myself. I would start with this and refine it for the company I was working for. Modifying and adjusting with iteration of use.

Graphical user interface, text, application, email

Description automatically generated

<https://www.securitymetrics.com/learn/penetration-testing-timeline-checklist>

**Create a planning and preparation checklist common to all security tests as a whole.**

For this one I found an indepth spreadsheet based checklist created by the DoD Environmental Research Program. They based their checklist off of the SANS approach. The credibility that those two organizations lends to me viewing their work better than my own. I would follow a similar route of taking aspects of this to iterate and refine within my own organization.

Table

Description automatically generated

<https://www.serdp-estcp.org/Tools-and-Training/Installation-Energy-and-Water/Cybersecurity/Templates-and-Checklists/Templates-and-Checklists-Files/FRCS-Pentest-Checklist>

**Identify any special planning and preparation needed for each test.**

Much of the planning that comes to mind is allocating resources to successfully fulfill the test without impacting existing delivery timelines. This is a whitebox test so the reliance on developer/operational insight will be one of the bigger impacting issues. To combat this, I would first focus on very clearly defining the scope. Scope creep will kill the planning timeline so fight it first. The second focus would be to clearly say what is needed when requesting developer time. This will server two goals. One it will give us the most efficient use of the developer time resources. Secondly, when the timeline slips, we can proactively shield ourselves by sharing previous communications and how directly we communicated.

**Identify the scope for each test and identify any special considerations that need to be addressed.**

Each of the test have their own unique considerations. So there are no one size fits all considerations. Take DNS. Understanding the unique topology of the organization will dictate how do we attempt to corrupt or subvert DNS. Is it internally hosted or externally managed. Is DNSSEC utilized? Etc, etc…. These considerations are not applicable to IoT or OS fingerprinting. So once a test is identified as needed then considerations for that test would be developed on the fly.

**Create a list of at least five testing activities for each audit.**

1. Port Scanning
2. Packet Sniffing
3. Process Flow Modeling
4. Denial of Service Profiling
5. OS Versioning

**Provide recommendations for securing the database that are unique to Oracle.**

An Oracle list was found online. Trusting the wisdom of the crowds I would base a lot of my recommendations on these checklist.

1. Enable Data Dictionary Protection.
2. Use Oracle Advanced Security to encrypt network traffic.
3. Lock down the Oracle Listener with a well formed password.
4. Enable Oracle Connection Manager to help filter connectons down to known web servers.
5. Use Oracle Net’s “Valide Node Checking” to help prevent DDOS attacks.

<https://docs.oracle.com/cd/B12037_01/network.101/b10773/checklis.htm>

**Provide recommendations for securing the database that are unique to MySQL.**

1. Delete the test user and database that is created during installation.
2. Ensure that each MySQL account has a password.
3. Avoid using the commandline to supply passwords as arguments.
4. Set the root account’s password.
5. Avoid granting Global Privileges.

<https://cloud.ibm.com/docs/database-tools?topic=database-tools-dbt-mysql-security>

**Provide recommendations for securing the database that are unique to SQL Server**

1. Isolate server using existing Microsoft firewall and networking directories.
2. Remove all other external dependencies .
3. Enable updates for SQL Server tools and applications.
4. Utilize Microsoft monitoring tools.
5. Run SQL Server as a local account, not as an administrator.

<https://www.dnsstuff.com/sql-server-security>

**Hands On Project 10-2**

**How will the scope be identified?**

Black box testing is often defined by the limitations of not knowing the inner workings of what it is that you are testing. In this case a focus on functionality testing will dictate the scope.

**What will indicate the end of a test?**

A successful accomplishment of the goal. Or existence of a hard timeline.

**What special skills or characteristics will be required from the assessor that are not as necessary in white box testing scenarios?**

The ability to perform network analysis and traffic analysis will be in higher demand. Much of what we take for granted during a white box test will only be reveiled externally as traffic fingerprinting.

**Identify and describe the first three main goals of the test.**

Reconnaissance, Access, and Elevate.

**Explain at least three specific techniques that will be used to gather information.**

Reconnaissance

1. Network monitoring.
2. Traffic analysis.
3. Network replay.

**Explain at least three specific techniques that will be used as an attempt to obtain access to the system.**

Access:

1. Default accounts and passwords
2. Unpatched CVE’s that allow for access.
3. Insecurely stored credentials.

**Provide at least two special considerations unique to Oracle.**

1. Attack the shared global area in memory.
2. Use established fuzzing tools and test patterns to Oracle.

**Provide at least two special considerations unique to MySQL.**

1. MySQL is notorious for creating passwordless accounts. Validate that
2. SchemaCrawler and other tools allow of users intuit the schema.

**Provide at least two special considerations unique to SQL Server.**

1. MS Flight Recorder will be invaluable for the remote casing. Gaining access to this is large lateral attack.
2. It too has many default accounts with no password. <https://www.securityfocus.com/bid/4797>