Becoming a Successful Pentester

Introduction

This article will cover the principles and steps required become a successful penetration tester. Penetration testing is often called “pentesting”. In the day and age of increasingly common data breaches and the resulting penalties and brand damage that can and will likely result, it has become a common practice to require pentesting as part of standard best practices in cybersecurity and compliance frameworks. The popularity of pentesting has greatly increased the demand for good pentesters, so much in fact that there are entire organizations that specialize in this practice. However, if your organization cannot afford to procure professional pentesting services, you may be forced to grow your pentesting team in-house, and that is why this article may be useful. This article could therefore be considered as Pentesting 101, however, because the area of security testing tools is rapidly changing and is in a constant state of flux, this article is not “tool centric”.

Becoming a Successful Pentester

In a field that is as important and rapidly changing as pentesting is, it is highly advisable that you collect and curate as much information as possible on the security-related tools that are available to stay abreast of the latest technologies and testing methods: In my own technical digital library of Security Tools, these are the sub-directories where I keep this information for learning and for reference when I collect it:

00_2017_Pentesting_Tools
00_2018_Pentesting_Tools
00_2019_Tools
00_Angry_IP_Scanner
00_Applied_Security_Visualization
00_Arachni
00_ASTo_for_IoT
00_Autosploit
00_AWS_Zeus
00_Barkley_Cybersecurity_Toolkit
00_Belati
00_Brutal_Kangaroo
00_BruteX
00_Burp
00_Cartero
00_CloudPiercer
00_CrackMapExec_Pentesting Active Directory Environments
00_Crowbar_Web Application Brute Force Attack
00_Cryptomator
00_CurrPorts
00_cve-search
00_CyberObserver
00_Detect_KRACK_Attacks
00_DNSDB
00_DNSDiag
00_dnsmap
00_Dripcap
00_DumpsterFire Toolset Security Incidents In A Box
00_Elite_Field_Kit_by_Hak5
00_Emergynt_Instinct_Engine

00_Eternal_Blue
00_FAME__Open Source Malware Analysis Platform
00_Fiddler
00_Gramfuzz
00_Hacking_Tools
00_HellRaiser_Vulnerability_Scanner
00_HijackThis
00_Incident Response Forensic Framework--NightHawk
00_IR_Rescue_Windows_Forensic_Data_Collection
00_Kai_Pfiester
00_Kali_Linux
00_Knowledge_Management
00_Kvasir_Penetration Test Data Management
00_LAN_Turtle
00_Linux_Distributions_for_Pentesting
00_LOIC--Low Orbit Ion Cannon
00_Lordix
00_Machinae__Security Intelligence Collector
00_Maltego
00_Mantra
00_Metasploit
00_Mitre_Attack_Test_Tools
00_Mobile_Security_Framework_MobSF
00_Morpheus_Ettercap
00_Netcat
00_Netcraft
00_Netsparkler
00_Nexpose
00_NextWare_Cyber_Collaboration_Toolkit
00_Nikto
00_Nishang_for_Powershell_Penetration_Testing
00_nmap
00_NOSQL_Exploitation_Framework
00_Open Source Firewall - OPNsense
00_OpenVAS
00_Open_Source_Network_Security_
00_Open_Source_Reconnaissance
00_Ostinato
00_OWASP Mutillidae_Web_App_Pentetsting
00_OWASP Offensive Web Testing Framework OWFT
00_p0f - Passive Traffic Analysis OS Fingerprinting and Forensics Tool
00_P4wnP1 highly customizable USB attack platform, based on a low cost Raspberry Pi Zero
00_Penetration_Testing_Tools
00_Pentesting
00_Pentest_Toolbox
00_Phishing
00_PirateBox
00_Powershell_Penetration_Testing_Framework
00_Privacy_Tools
00_ProcDOT

00_PRTG_Network_Monitor
00_PTF
00_PUTTY
00_PWNIE_EXPRESS
00_PwnPad
00_Quad9
00_ReconScan
00_Recovery Boot Password Reset
00_Red_Team
00_RevIP__Reverse IP Lookup Tool
00_RF_Hacking_Field_Kit
00_scanless_Public_Port_Scan_Scrapper
00_SCP
00_Security_Onion
00_Shodan
00_Slowloris
00_Sparta_Vulnerability_Scanner
00_Sploitego
00_Splunk
00_Spyware_Removal
00_SQLiv
00_SQLMap
00_Stackhackr
00_TCPDump
00_TDSSKiller
00_The_Harvester
00_Tools_Watch
00_Top Best Ethical Hacking Tools 2018
00_Tor
00_Tor_Browser
00_USB_Amory
00_USB_Canary
00_USB_Rubber_Ducky
00_v3n0m
00_vane_- WordPress Vulnerability Scanner
00_W3AF
00_WATOBO
00_WebGoat
00_WebPwn3r
00_Wfuzz
00_WiFite_Automated_Wireless_Attack
00_Wifi_Pineapple
00_Windows_Warez
00_WinDump
00_Wireless_Gear_by_Hak5
00_Wireshark
00_WPForce - Wordpress Attack Suite
00_WS-Attacker
00_Yeti
00_ZAP

Becoming a Successful Pentester

Why is Pentesting Important?

Pentesting is important primarily for two major reasons: 1) You want to identify (and ultimately remediate) your digital infrastructure weaknesses before the bad guys find and exploit them; and 2) many major cybersecurity compliance frameworks, such as PCI DSS and SOC 2 Type 2, and the New York Department of Financial Services Cybersecurity Regulation require periodic pentesting in order to achieve the minimum required level of compliance. Failure to achieve compliance can have serious repercussions to the well-being of an organization, its leadership, and its stakeholders.

Your Goals in Pentesting

Your goals pentesting should be to:

1) Satisfy your Stakeholders' business requirements to make the organization more secure
2) Find and document your infrastructure and application vulnerabilities before the bad guys do
3) Choose the right tools that are going to reveal the best results in the shortest time possible
4) Document the required remediations
5) Report the results to your management
6) Do all this as efficiently and effectively as possible, in the shortest amount time possible

What Do You Have to Know?

Obviously, you need to a fair amount about the system and tools you are using, as well as the system or systems on which you are performing the pentesting. But if you are just starting out and aspire to be a great pentester, then this article and the resources listed at the end, are a good starting point. Knowing how to search websites using Google can also shorten your learning curve. The important thing to understand is that the landscape is constantly changing. If you follow this path, as in most areas of cybersecurity, you will have to dedicate yourself to being in a constant state of research and learning. What was important 12 or 18 months ago may not be so important now. Likewise, the tools associated with automated testing, such as ZAP and Netsparker, continue to improve and get more powerful, so to give yourself an edge and to be competitive it is important to understand how to use these tools, and to understand the results they produce in their reports.

The Basics

Black Box Pentesting or White Box Pentesting?

In order to assure that every area of your infrastructure security is reliable and effective, penetration tests should be conducted on a regular basis, at least every six months, and probably every three months, if your stakeholders will permit it. As deficiencies are noted, corrections and improvements should be made. There are two general types of audits, the Black Box Pentests and the White Box Pentests.

Black Box Pentesting

In a black-box pentest the pentester is only provided with and IP address or a range of IP addresses to scan and probe for known vulnerabilities, much the same as a hacker would. Sometimes this is known as vulnerability scanning or penetration testing. Advantages of Black-Box audits include that fact that they are faster, cheaper and simpler than White-Box Audits. Disadvantages of Black-Box Audits include the fact that they will not uncover configuration errors, errors in policies and procedures, and errors in design (Norberg, 2001). See Figure 1 for a diagram of a Black-Box Pentest.

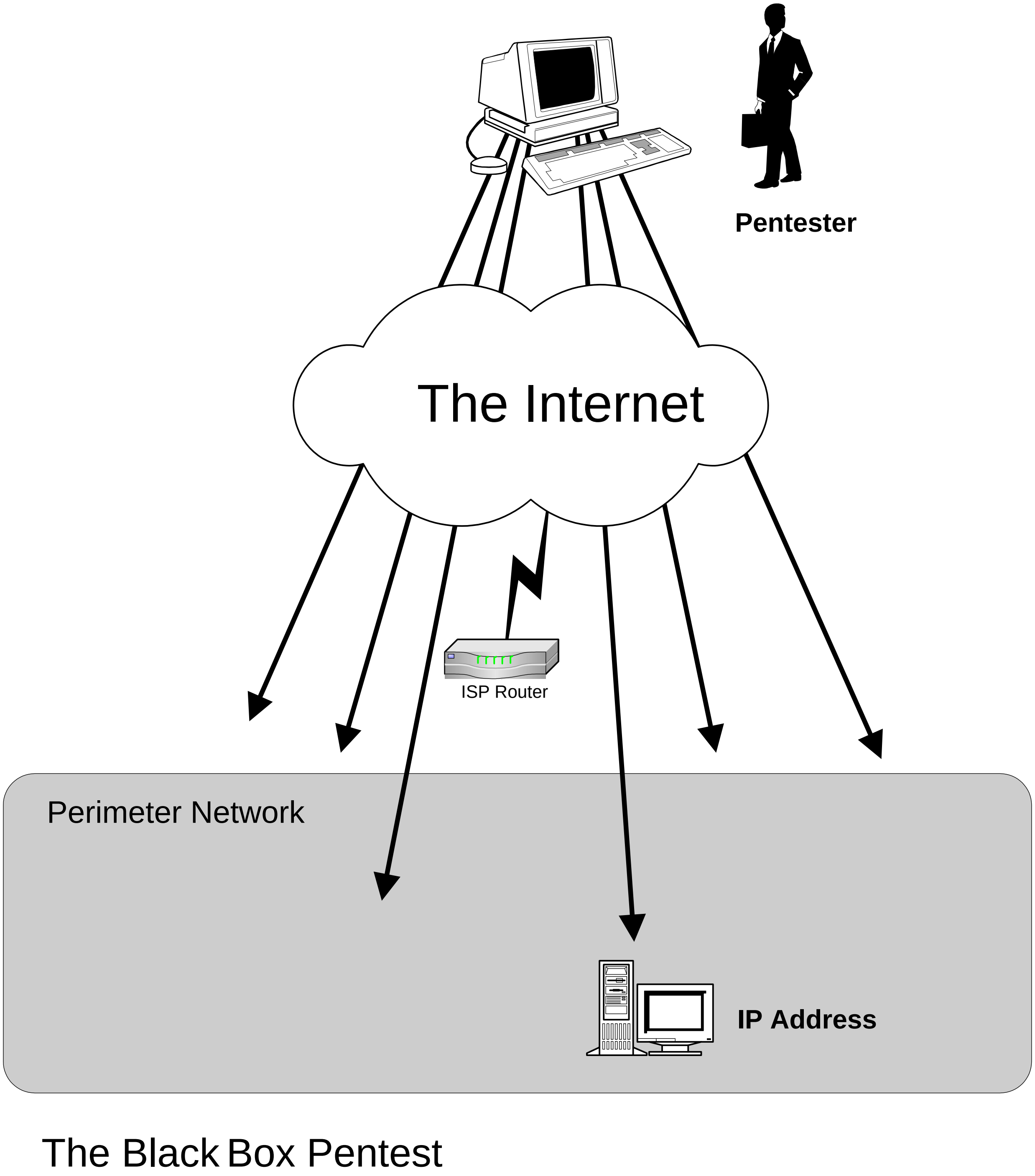


Figure 1 -- Black-Box Pentest (Norberg, 2001)

White Box Pentest

As shown in Figure 2 below, the White-Box Pentest will involve auditors working inside your site. “They will require all possible information about your site, including network diagrams, configuration files, all available documentation of the systems. Using all this information, the pentesters will be able to identify possible theoretical attacks against your environment. The pentesters should also review and comment on your policy documents, for example your backup policy (Norberg, 2001).” Norberg recommends the White-Box approach over the Black-Box approach, despite its additional consumption of resources and additional length of time, because it will pay off in terms of finding more problem areas and vulnerabilities. The end result, if you follow through on their recommendations will be a more secure facility and IT resources.

Note: The three known disadvantages of White Box Pentesting are 1) it takes usually takes longer; 2) it is an unrealistic view of the infrastructure because a hacker will actually have limited knowledge of the infrastructure; and 3) the quality of the pentest results will be closely tied to the accuracy and completeness of the infrastructure documentation provided. Many organizations don’t do a very good job of keeping their infrastructure documentation updated and complete, so the quality of the pentest project will not be good if the documentation provided is not good quality. Remember, your time is limited, and the scope of your job is to find serious technical vulnerabilities as quickly as possible, not to be responsible for the accuracy or completeness of your client’s infrastructure documentation.

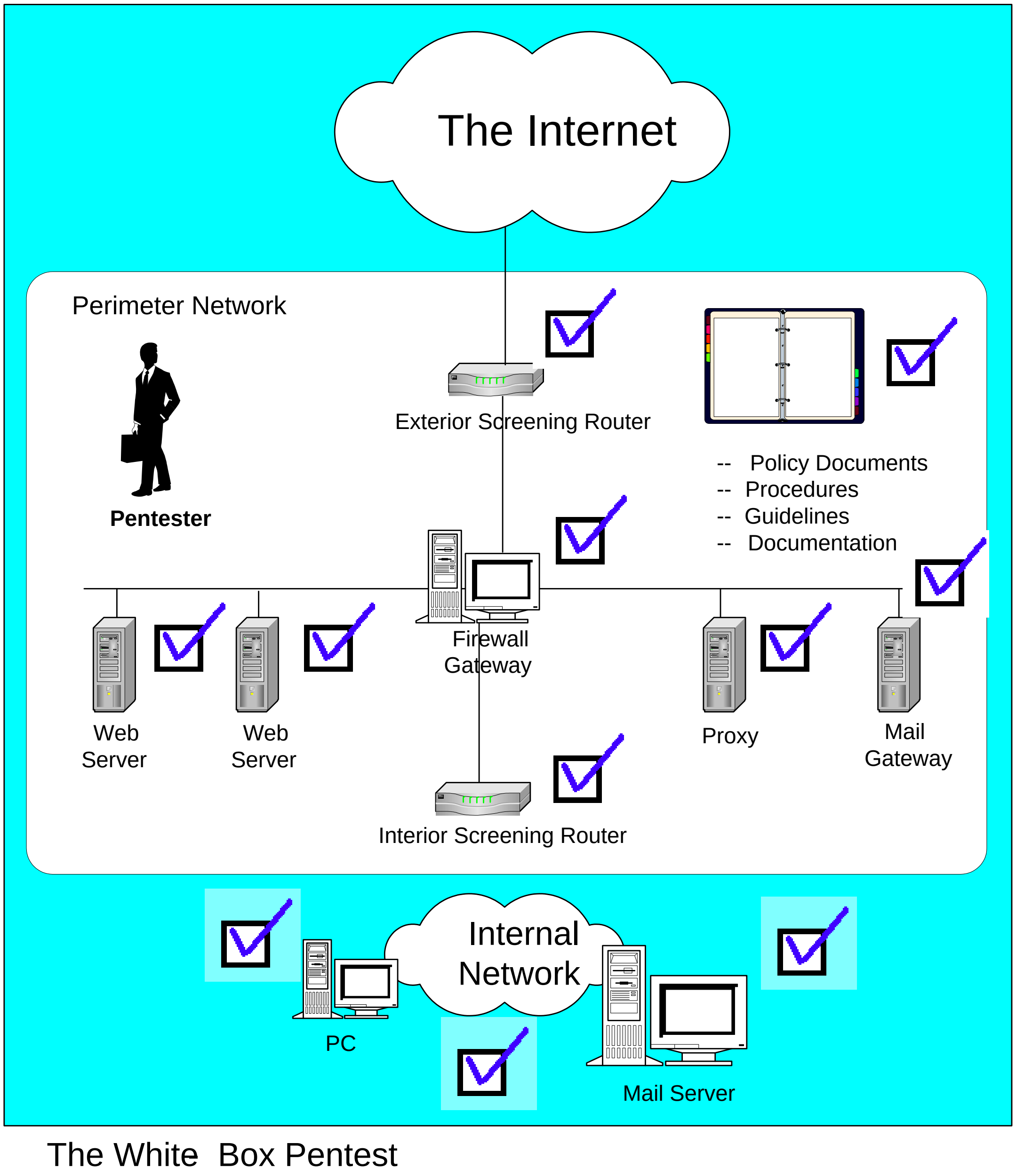


Figure 2 -- White-Box Pentesting (Norberg, 2001)

Choose or Adopt a Methodology

Often you will be asked if you have a methodology for your pentesting. There are several structured approaches, but here are three methodologies that you can explore and possibly select and use. The Information System Security Assessment Framework (ISSAF), the Open Source Security Testing Methodology Manual (OSSTMM), or a simplistic, structured approach which I developed to get the job done fast. Your stakeholders and/or project sponsor may favor one over another, so it is good to be knowledgeable on each of these and to be flexible, so you can be more well-rounded, and just in case you need this information for a job interview in the future. General details of each of these methodologies are covered below.

ISSAF

The ISSAF pentesting is based on the Project Management Institute's Project Management Body of Knowledge standard methodology. If you are a Project Management Professional (PMP), you may be very comfortable with this methodology. These are the pentesting assessment phases under ISSAF:

- Planning and Preparation
- Assessment
  - Information Gathering
  - Network Mapping
  - Vulnerability Identification
  - Penetration
  - Gaining Access and Privilege Escalation
  - Enumerating Further
  - Compromise Remote (Users and Sites)
  - Maintaining Access
  - Covering Tracks
- Network Security
  - Password Security Assessment
  - Switch Security Assessment
  - Router Security Assessment
  - Intrusion Detection System Security Assessment
  - Virtual Private Network Security Assessment
  - Antivirus System Security Assessment and Management Strategy
  - Storage Area Network Security Assessment
  - Internet User Security
  - E-Mail Security
- Host Security
- Application Security
- Database Security
- Social Engineering
- Reporting
  - Reporting
  - Clean-up and Destroy Artifacts

OSSTMM

The OSSTMM was created by the Institute for Security and Open Methodologies (ISECOM) and the current version is OSSTMM 3.0, and it is approximately 213 pages in length. You can obtain the OSSTMM for free here: http://www.isecom.org/research/. Note: OSSTMM 4.0 is in draft format. This is the structure of the OSSTMM 3.0.

- Rules of Engagement
- Channels
  - Network Security
  - Physical Security

    - Wireless Communications
    - Telecommunications
    - Data Networks
        - Network Surveying
        - Enumeration
        - Identification
        - Access Process
        - Services Identification
        - Authentication
        - Spoofing
        - Phishing
        - Resource Abuse
- Modules (OSSTMM has repeatable processes in the form of Modules)
    - Phase I – Regulatory
        - Posture Review
        - Logistics
        - Active Detection Verification
    - Phase II – Definitions
        - Visibility
        - Access Verification
        - Trust Verification
        - Controls Verification
    - Phase III – Information Phase
        - Process Verification
        - Configuration Verification
        - Property Validation
        - Segregation Review
        - Exposure Verification
        - Competitive Intelligence Scouting
    - Phase IV – Interactive Controls Test Phase
        - Quantitative Verification
        - Privilege Audit
        - Survivability Validation
        - Alert and Log Review

Simplistic, Structured Approach

In my simplistic structured approach, I list 11 major tasks, each of which is described below.

Task 1

Collect details about Pentesting: Black Box or White Box, Goals, etc.

Task 2

Align pentesting goals to capabilities of pentesting tools to ensure the necessary tools are available. If unavailable, obtain the necessary pentesting tools.

Task 3

Create prospective schedule and get permission from Management and if applicable, the Cloud Services Provider (i.e. Amazon Web Services) to perform the pentesting

Task 4
Create the pentesting project plan

Task 5
Communicate the pentesting start time prior to commencing the pentesting.

Task 6
Start the pentesting.

Task 6
Check the Pentesting Results and Reports

Task 7
Communicate the completion of pentesting to the Stakeholders

Task 8
Check the Pentesting Results. Organize, Label and Analyze the Results

Task 9
Create the first draft of the Report to show the Pentesting Results, Analysis and Recommendations

Task 10
Deliver the first draft of the Pentesting Results Report. Revise as required,

Task 11
Prepare and distribute Pentesting Results Report to the Project Stakeholders.

Write a Detailed Plan

No pentesting project should take place without an organized plan to describe what you are doing, and when, where, and how you will do it.   Besides ensuring that such a professional approach will increase stakeholder confidence that you know what you are doing, preparing a plan like this will help keep you on track and it will make the final pentesting project results report much easier to write. Special note: If you are a contractor, both the pentesting project plan and the pentesting results report will be contractually required deliverables. Fail to produce these and you will likely not be paid nor invited back for return engagements. These are the steps to writing the pentesting project plan:

Communicate:

What & why you will do it
When you will do it
How you will do it
Where you will do it
Create and provide a general attack diagram to communicate to stakeholders what you are doing

Get your pentesting tools ready and ensure that you know very well how to use them

Get Permission from your Stakeholders

Get the Plan approved

Follow the Plan

Figure 3 below shows an example decision flow chart from one of my pentest project plans

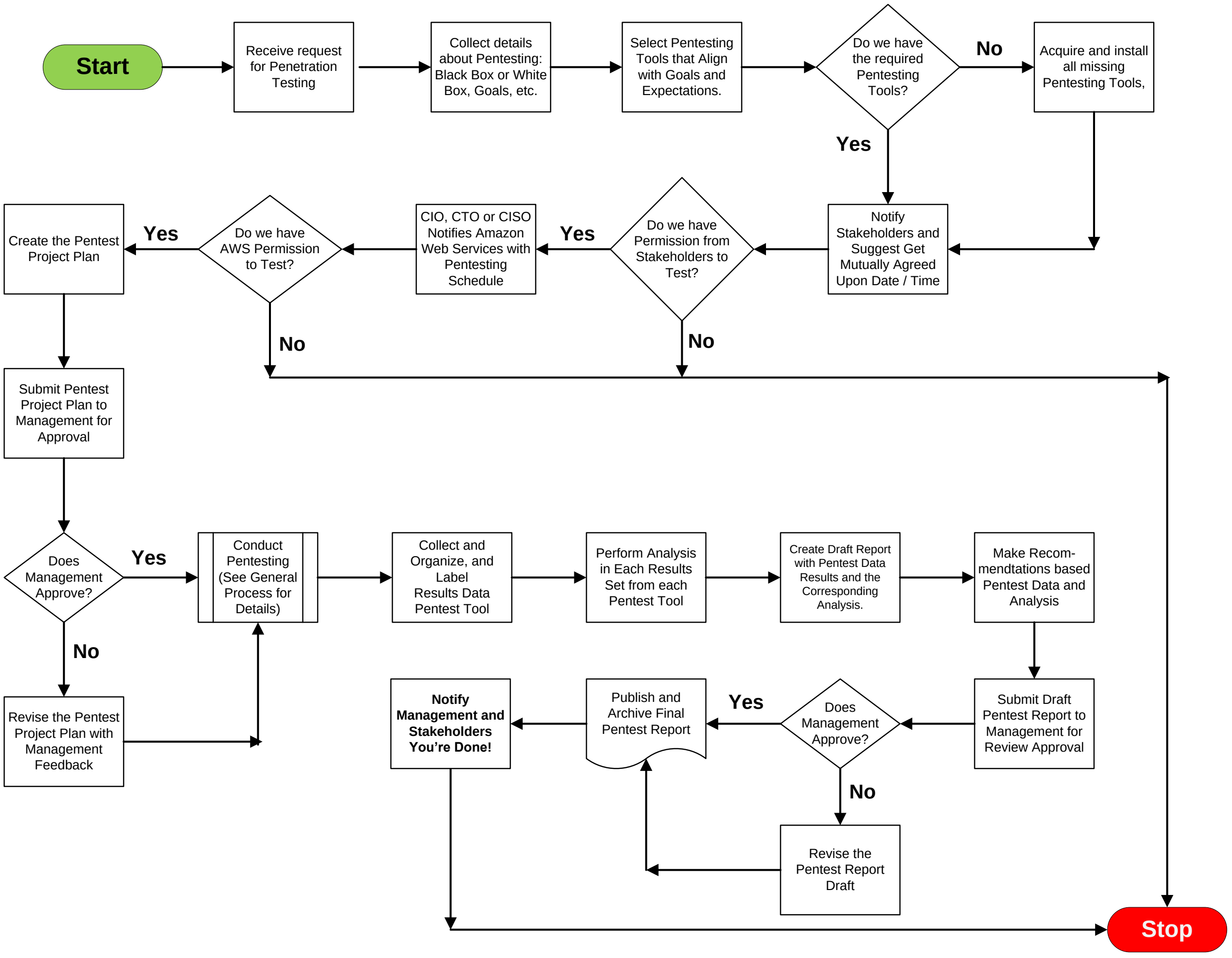

Start
Receive request for Penetration Testing
Collect details about Pentesting: Black Box or White Box, Goals, etc.
Select Pentesting Tools that Align with Goals and Expectations.
Do we have the required Pentesting Tools?
No
Acquire and install all missing Pentesting Tools,
Yes
Notify Stakeholders and Suggest Get Mutually Agreed Upon Date / Time
Do we have Permission from Stakeholders to Test?
Yes
CIO, CTO or CISO Notifies Amazon Web Services with Pentesting Schedule
Do we have AWS Permission to Test?
Yes
Create the Pentest Project Plan
No
No
Submit Pentest Project Plan to Management for Approval
Does Management Approve?
Yes
Conduct Pentesting (See General Process for Details)
Collect and Organize, and Label Results Data Pentest Tool
Perform Analysis in Each Results Set from each Pentest Tool
Create Draft Report with Pentest Data Results and the Corresponding Analysis.
Make Recom-mendtations based Pentest Data and Analysis
No
Revise the Pentest Project Plan with Management Feedback
Submit Draft Pentest Report to Management for Review Approval
Does Management Approve?
Yes
Publish and Archive Final Pentest Report
Notify Management and Stakeholders You're Done!
No
Revise the Pentest Report Draft
Stop

Figure 3 shows an example high-level pentesting schedule diagram

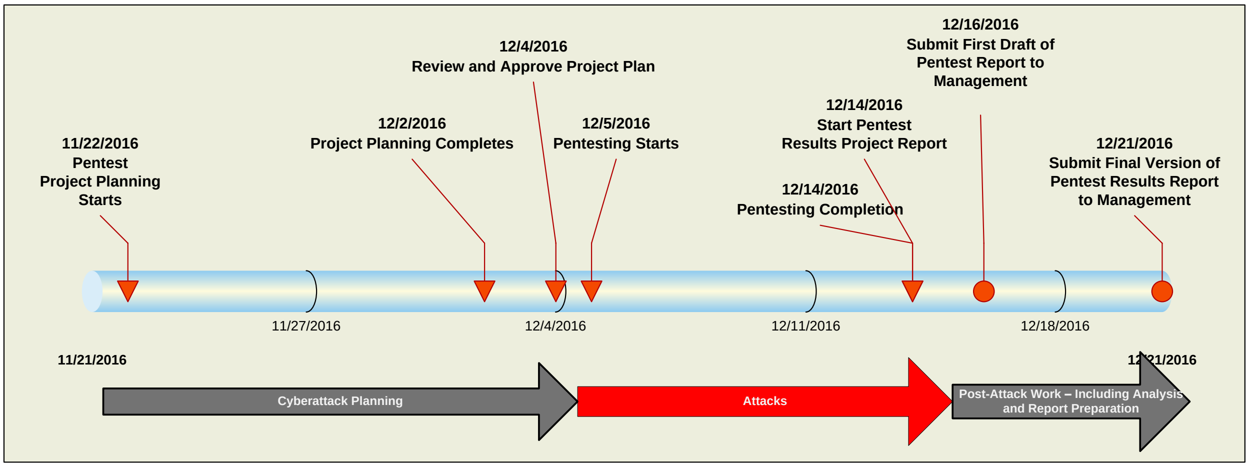


Figure 3 – High-Level Pentesting Schedule

Figure 4 below shows an example high-level attack diagram from one of my pentest project plans. The Black Box Penetration testing was performed in phases as show in the annotated diagram below, beginning from the Preparation Phase and attempting to progress as far as possible, without destroying data.

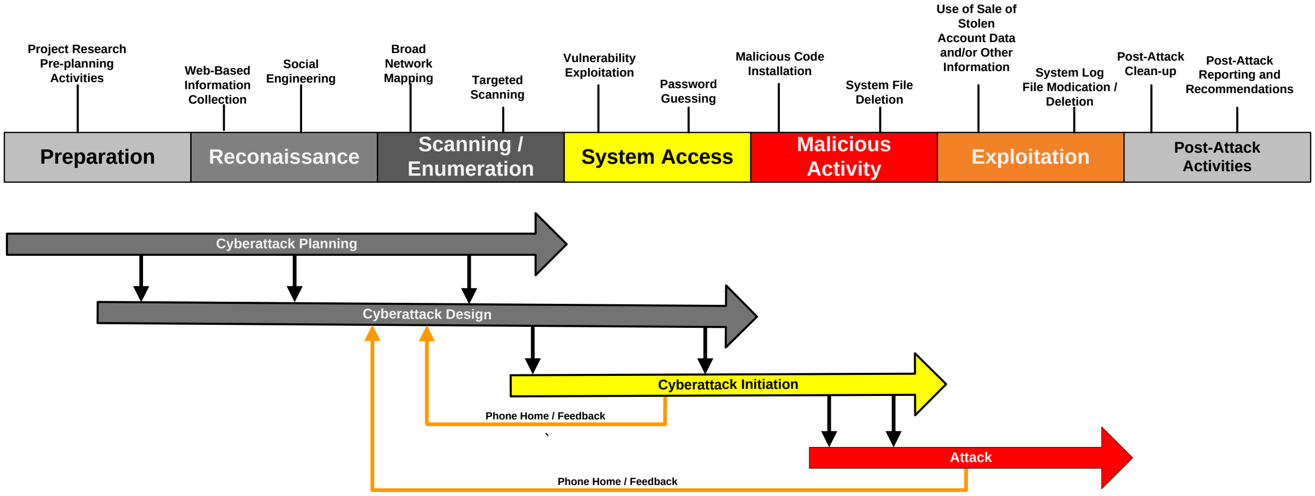


Figure 4 - High-level Attack Diagram

Figure 5 below shows high-level infrastructure diagram that was depicting the Black Box pentesting that would be taking place. Special note: In the Amazon Web Services Model, they did allow Internet traffic in for my pre-coordinated IP address, however, only on TCP ports 23, 80, and 443.

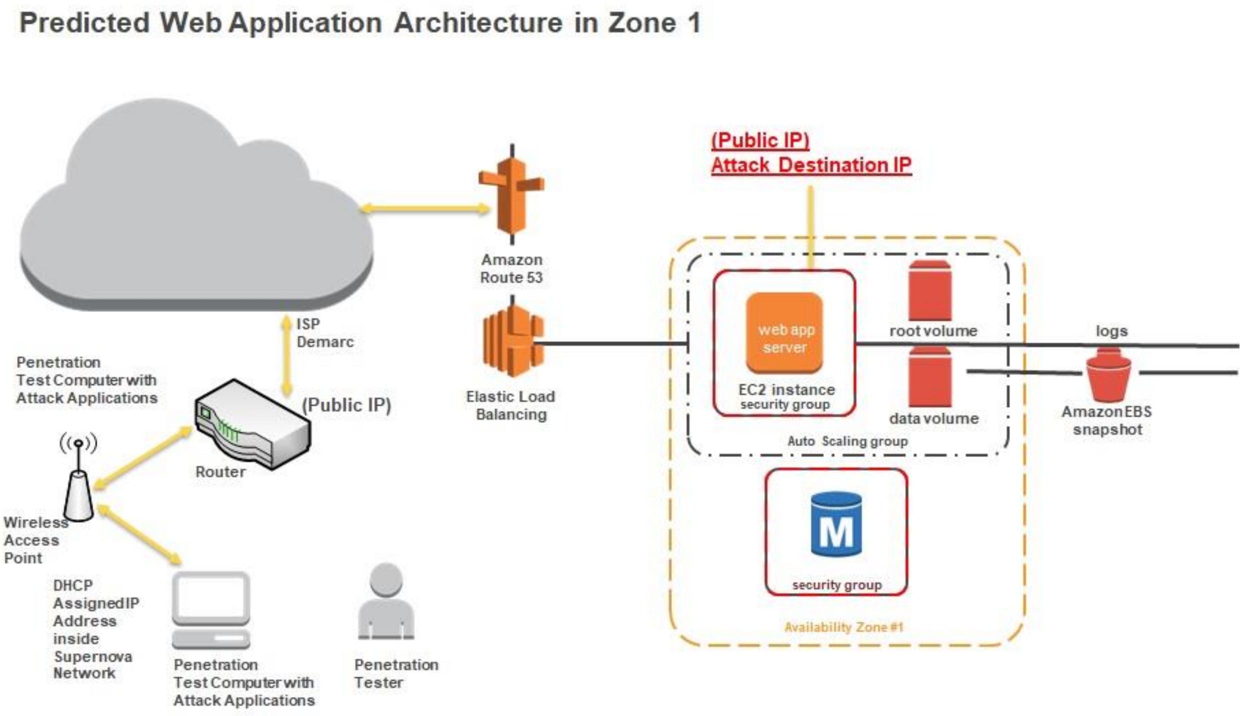


Figure 5 - High-level infrastructure diagram that was depicting the Black Box pentesting that would be taking place

Figure 6 below shows the Table of Contents for one of my Pentesting Project Plans.

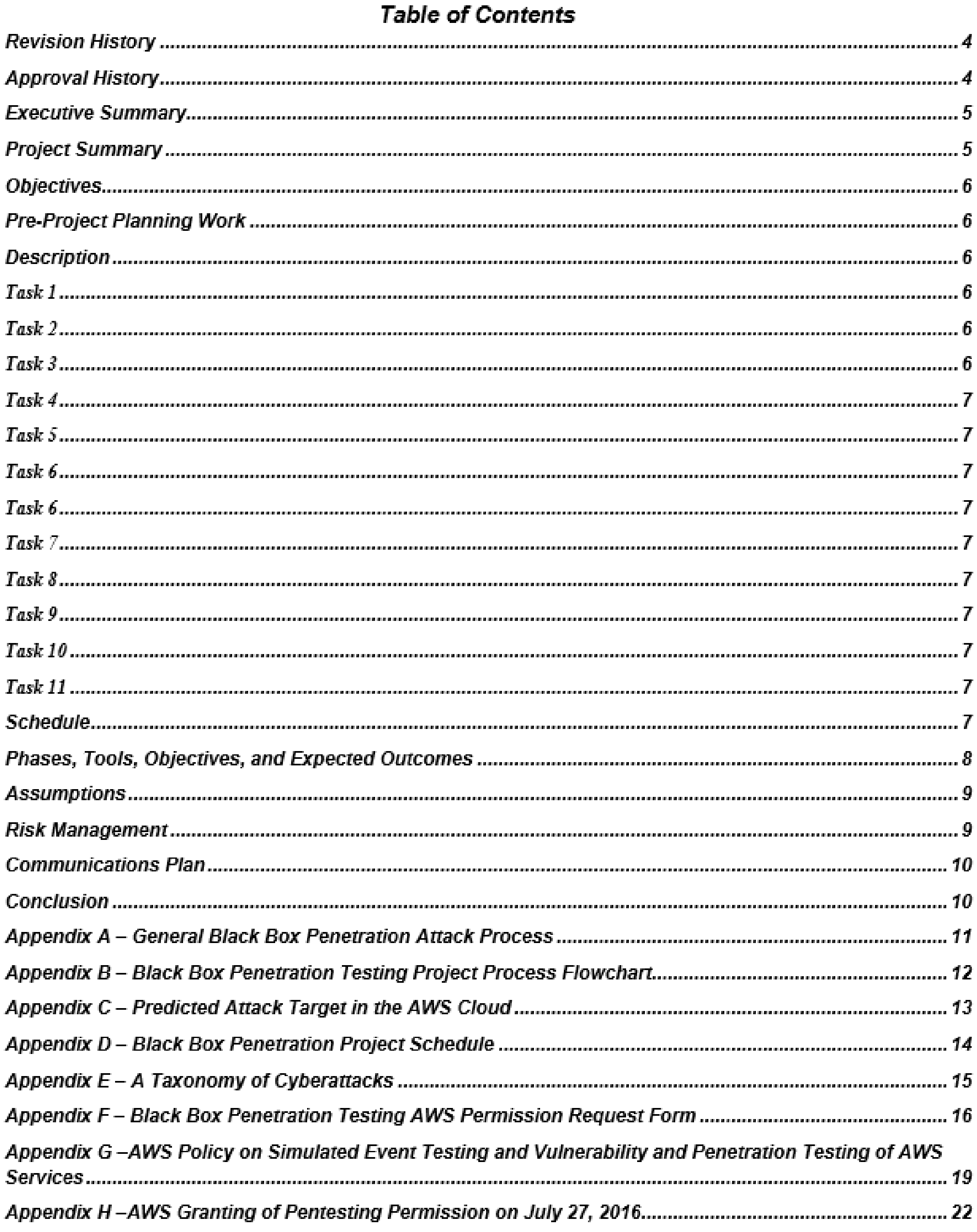


Table of Contents

Figure 6 - Table of Contents for one of my Pentesting Project Plans.

Prepare your Attack Machine and Tools

It is important to understand that your pentesting attack machine will not only have to have its pentesting tools installed and configured, but it will also have to have ALL of its security defense software either disabled or uninstalled. This is because many of the security-related tools will kill and/or disable your attack tools.

Start the Testing

Here are some important points to remember when you begin your pentesting.

- Follow the Pentest Project Plan
- Carefully collect, name, and organize and name all the artifacts in a structured way. Example:
    - **Black_Box_Pentesting_Data_Day_01_Test_02_from_William_Slater_2016_0801**
- Pace yourself and be methodical and thorough
- If something doesn't work as you expect, research and ensure that you are doing everything correctly.
- Time IS NOT your friend.
- Always tell the truth about your findings, and what worked and what didn't
- Be ready at any point to give a clear status report (or presentation) of your progress at a moment's notice. Your stakeholders will likely want to understand what your progress is and understand that you know what you are doing.

Write the Pentest Report

Up to this point, if you did your pentest project plan and saved your artifacts as recommended, your pentest report should be pretty simple to write. Just ensure that you are thorough, organized, and truthful when you write the report. Figure 7 below

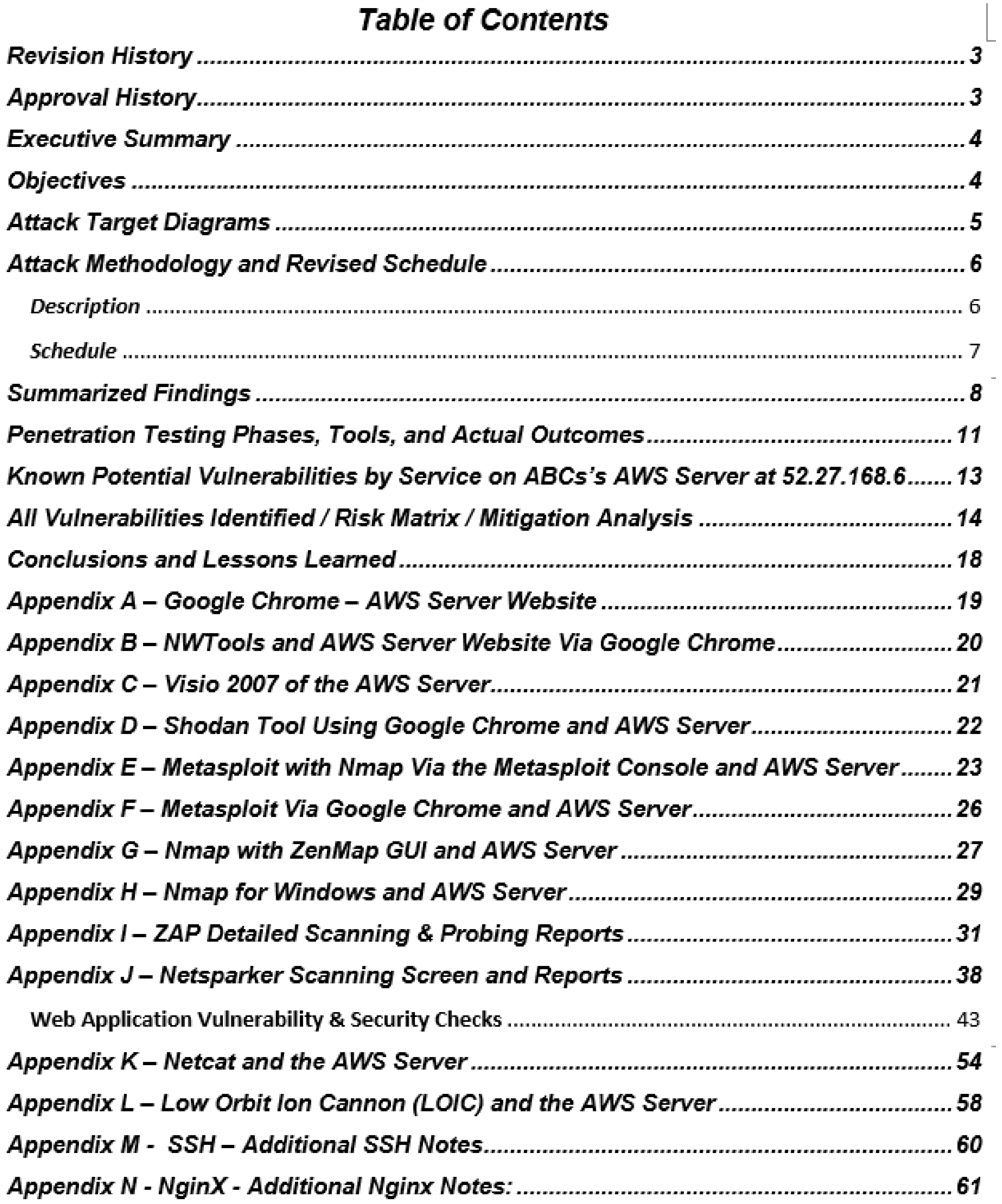


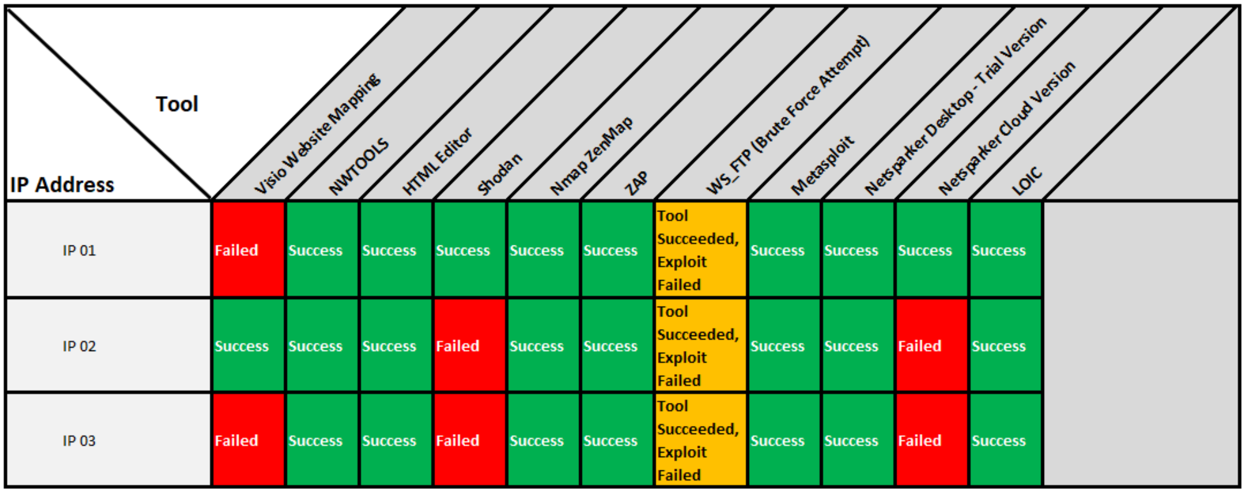
Table of Contents

Figure 7 – Example Table of Contents for the Pentesting Results Report

Becoming a Successful Pentester

Example Summary

In Figure 8 below, you can see an example summary report of pentesting tools and results. My Stakeholders appreciated this high-level summary.



Tool IP Address	Visio Website Mapping	NWTOOLS	HTML Editor	Shodan	Nmap ZenMap	ZAP	WS_FTP (Brute Force Attempt)	Metasploit	Netsparker Desktop - Trial Version	Netsparker Cloud Version	LOIC
IP 01	Failed	Success	Success	Success	Success	Success	Tool Succeeded, Exploit Failed	Success	Success	Success	Success
IP 02	Success	Success	Success	Failed	Success	Success	Tool Succeeded, Exploit Failed	Success	Success	Failed	Success
IP 03	Failed	Success	Success	Failed	Success	Success	Tool Succeeded, Exploit Failed	Success	Success	Failed	Success

Figure 8 – Summary of Pentesting Results

Remediate the Results

After you complete the pentesting results report, you need to either work with those responsible for the remediation of the vulnerabilities you identify to ensure that all those vulnerabilities get remediated, according to their severity, in as a timely manner as possible. There are two reasons for this: 1) every vulnerability identified is an opportunity for a bad guy to perform a successful exploit; and 2) any and every IT Security Auditor will require evidence that each of these identified vulnerabilities were remediated in a timely fashion. To do anything less is to risk busting an audit.

Conclusion

Pentesting is not "Rocket Science" and if you follow the advice and steps in this article, you can and will be successful. Remember that being structured and organized, and knowing your tools are your keys to success.

Note: In 1999, I wrote an article that was published about web development using FrontPage 2000. For about 13 years afterwards, I received many polite e-mails requesting for information or asking for answers to additional questions. If you have questions or need assistance after reading this article, please e-mail me at slater@billslater.com and I will help you if I can.

References and Resources

Aber, R. (2003). A Comprehensive Approach to Security. [Electronic version]. Retrieved from the web on May 1, 2004 at http://www.bcr.com/bcrmag/2003/05/p17.asp.

Allen, J. H. (2001). The CERT Guide to System and Network Security Practices. Boston, MA: Addison-Wesley.

Englebretson, P. (2013). The Basics of Hacking and Penetration Testing, second edition. Boston, MA: Syngress.

ISECOM. (2018). OSSTMM 3.0 retrieved from http://www.isecom.org/mirror/OSSTMM.3.pdf.

Kaeo, M. (2004). Designing Network Security, second edition. Indianapolis, IN: Cisco Press.

Noonan, W. J. (2004). Hardening Network Infrastructure. Emeryville, CA: Osborne McGraw-Hill.

Norberg, S. (2001). Securing Windows 2000 Servers for the Internet. Beijing, China: O'Reilly & Associates.

Pfleeger, C. P. and Pfleeger, S. L. (2003). Security in Computing, Third Edition. Upper Saddle River, NJ: Prentice Hall.

Stallings, W. (2012). Network Security Essentials: Applications and Standards. Upper Saddle River, NJ: Prentice Hall.

Wilhelm, T. (2010). Professional Penetration Testing, Boston, MA: Syngress.

Byline:

William Favre Slater, III is a Sr. IT Consultant, Project Manager / Program Manager, and author in Cybersecurity, Blockchain, Data Centers, and several other important areas of Information Technology. His professional career began in the United States Air Force, and has more than three decades of experience in IT. He has three graduate degrees, one of which is in Cybersecurity, and 80 professional certifications. He lives and works in Chicago, and is the proud, devoted of husband of his lovely wife and best friend, Joanna Roguska.