SAMPLE NETWORK INFRASTRUCTURE

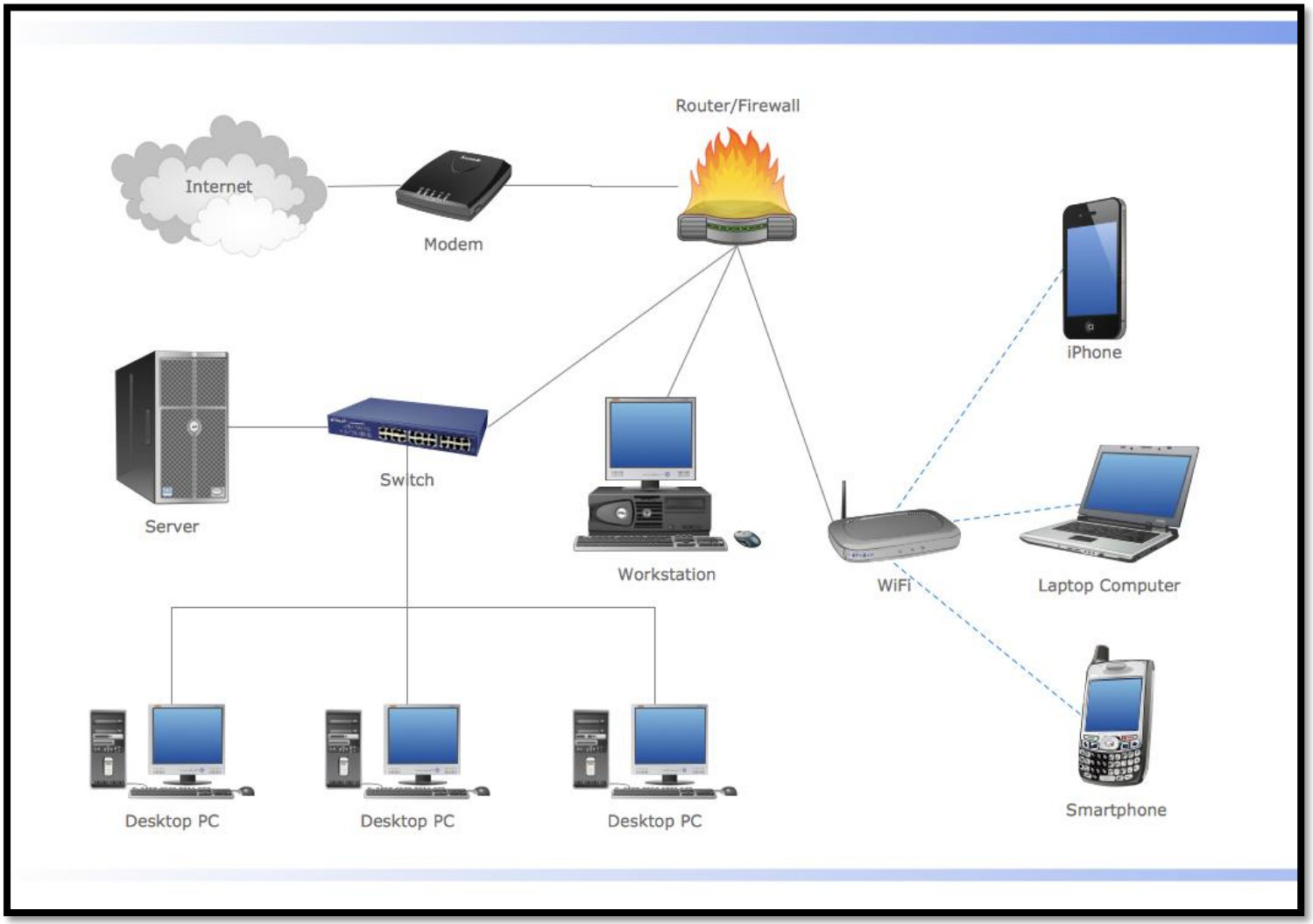
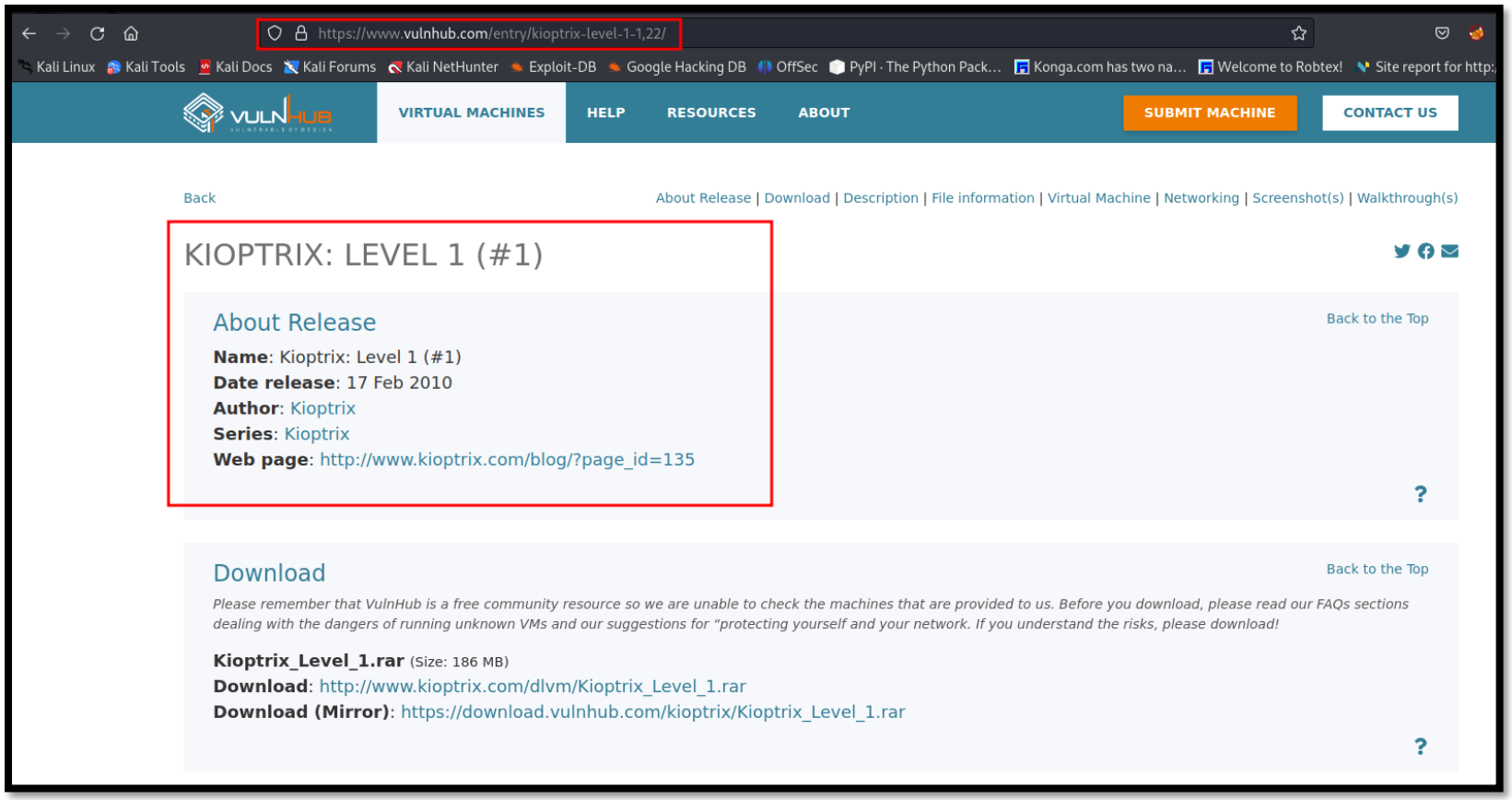


Photo credit : conceptdraw.com

Machine to scan against

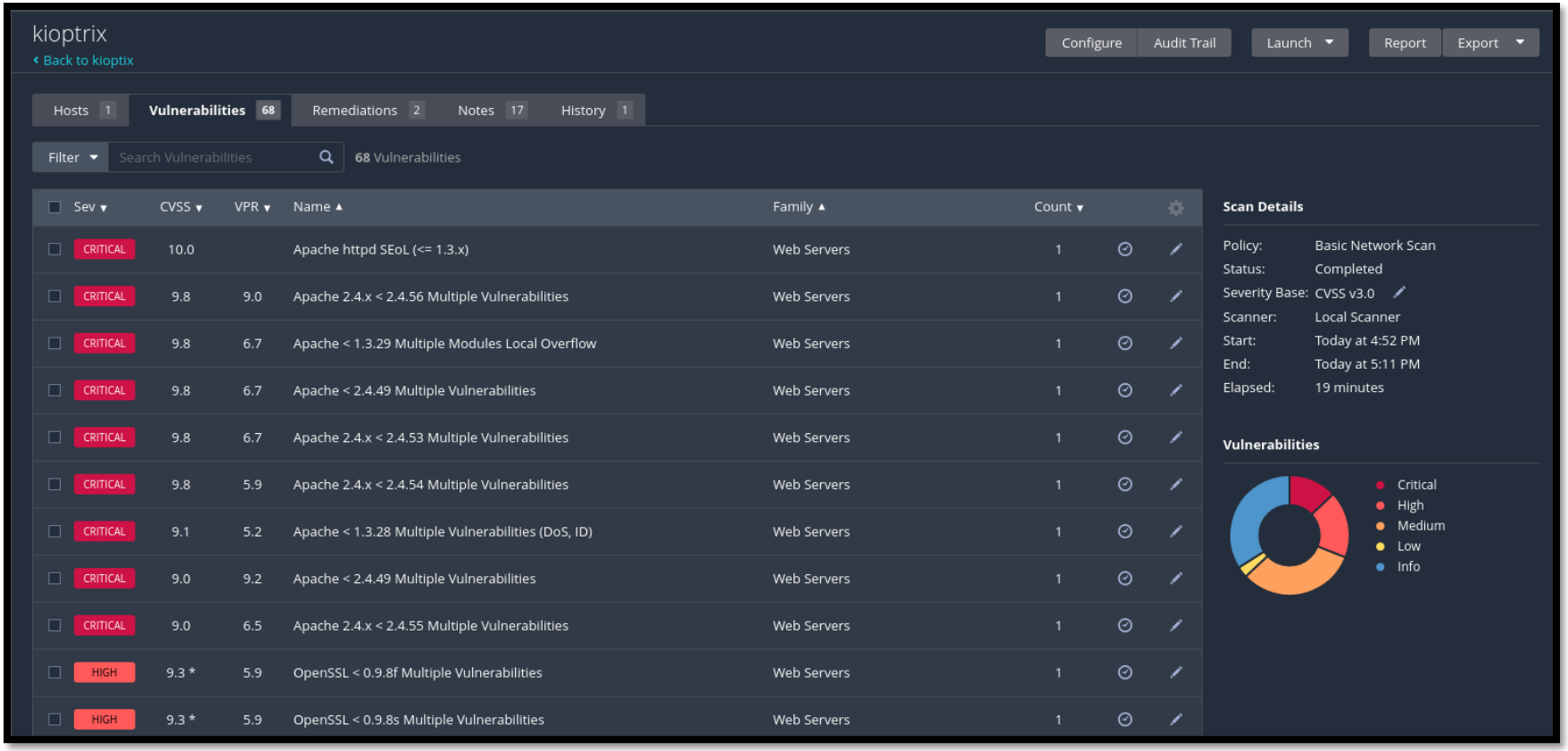


Kioptrix is a downloadable VM image file on Vulnhub. It is a VM image challenge to get root access by any means possible. The goal of these is to learn the basic tools and techniques in vulnerability assessment and exploitation.

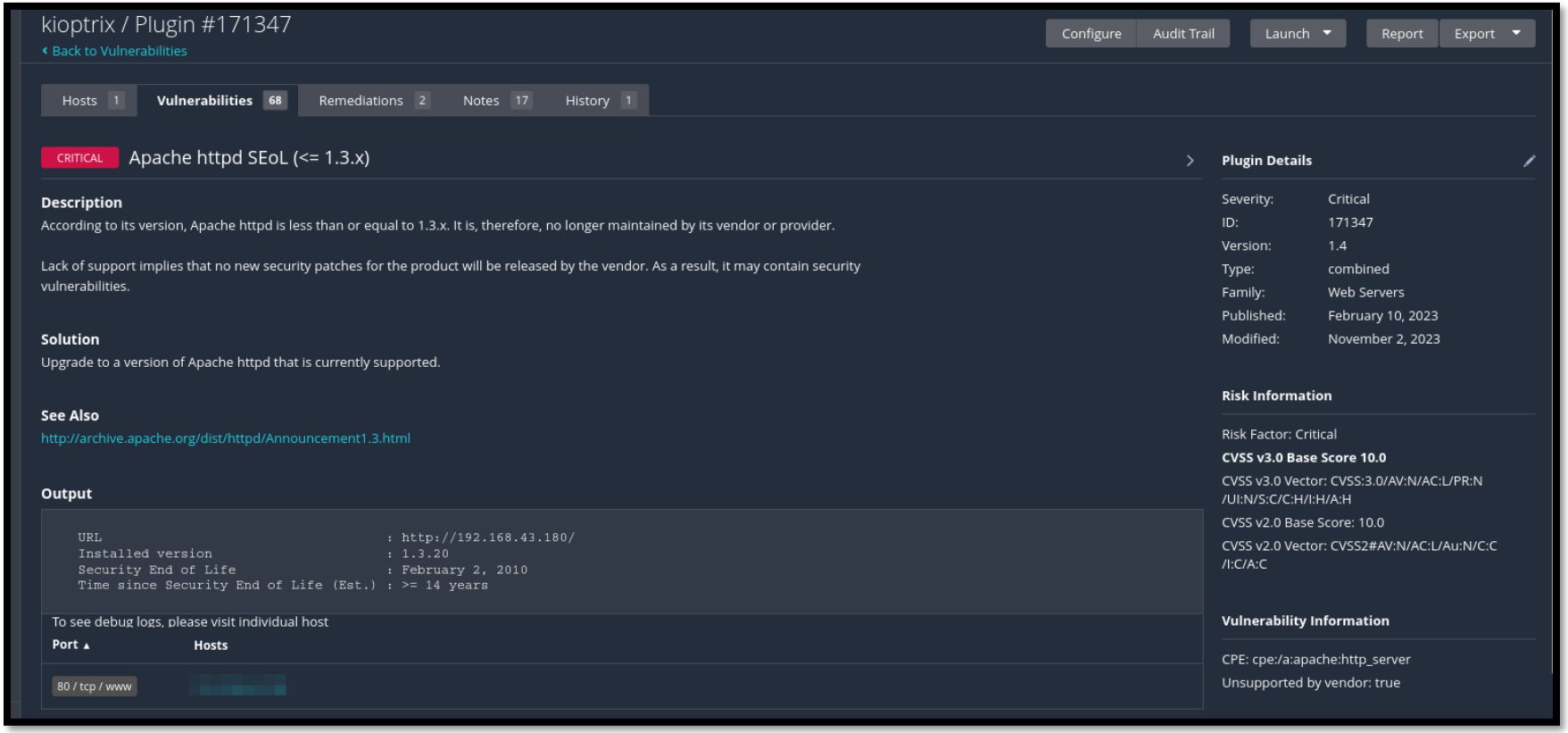
Scanning machine with Nessus vulnerability scanner

Nessus is a remote security scanning tool, which scans a computer and raises an alert if it discovers any vulnerabilities that malicious hackers could use to gain access to any computer you have connected to a network.

Vulnerabilities discovered with Nessus Scanner



Apache httpd SEoL (<= 1.3.x)



Mitigation for Apache httpd SEoL (<= 1.3.x):

Step 1: Identify Affected Systems

- Use vulnerability scanners or manual inspection to identify all systems running Apache httpd version 1.3.x.

Step 2: Verify Backup

- Before making any changes, ensure that you have a recent and verified backup of all critical data and configurations.

Step 3: Upgrade Apache httpd

- Download the latest version of Apache httpd (preferably version x.x.x or later) from the official Apache website.
- Follow the installation instructions provided with the Apache httpd package to install the new version.
- Ensure that all necessary dependencies are met and any custom configurations are migrated appropriately.

Step 4: Configuration Migration

- Review your existing Apache httpd configurations and ensure they are compatible with the new version.
- Make necessary adjustments to the configurations to ensure they work seamlessly with Apache httpd 2.4.x or later.
- Pay special attention to any deprecated or removed directives.

Step 5: Testing

- Test the new Apache httpd installation thoroughly in a non-production environment to ensure it functions correctly.
- Test various use cases, including serving web pages, handling different types of requests, and any custom configurations or modules.

Step 6: Deployment

- Once testing is successful, schedule a maintenance window to deploy the new Apache httpd version to production systems.
- Follow your organization's change management procedures for deploying changes to production systems.
- Monitor the deployment process closely to identify and address any issues promptly.

Step 7: Post-Deployment Testing

- After deployment, conduct thorough testing in the production environment to ensure that the new Apache httpd version is functioning as expected.
- Monitor system performance, error logs, and security logs for any abnormalities.
- Address any issues that arise during post-deployment testing promptly.

Estimated Timelines:

- The time required for mitigation depends on the complexity of your environment, the number of systems running Apache httpd 1.3.x, and the availability of resources.
- Planning and preparation: 1-2 weeks
- Testing and deployment: 1-2 weeks
- Post-deployment testing and monitoring: Ongoing

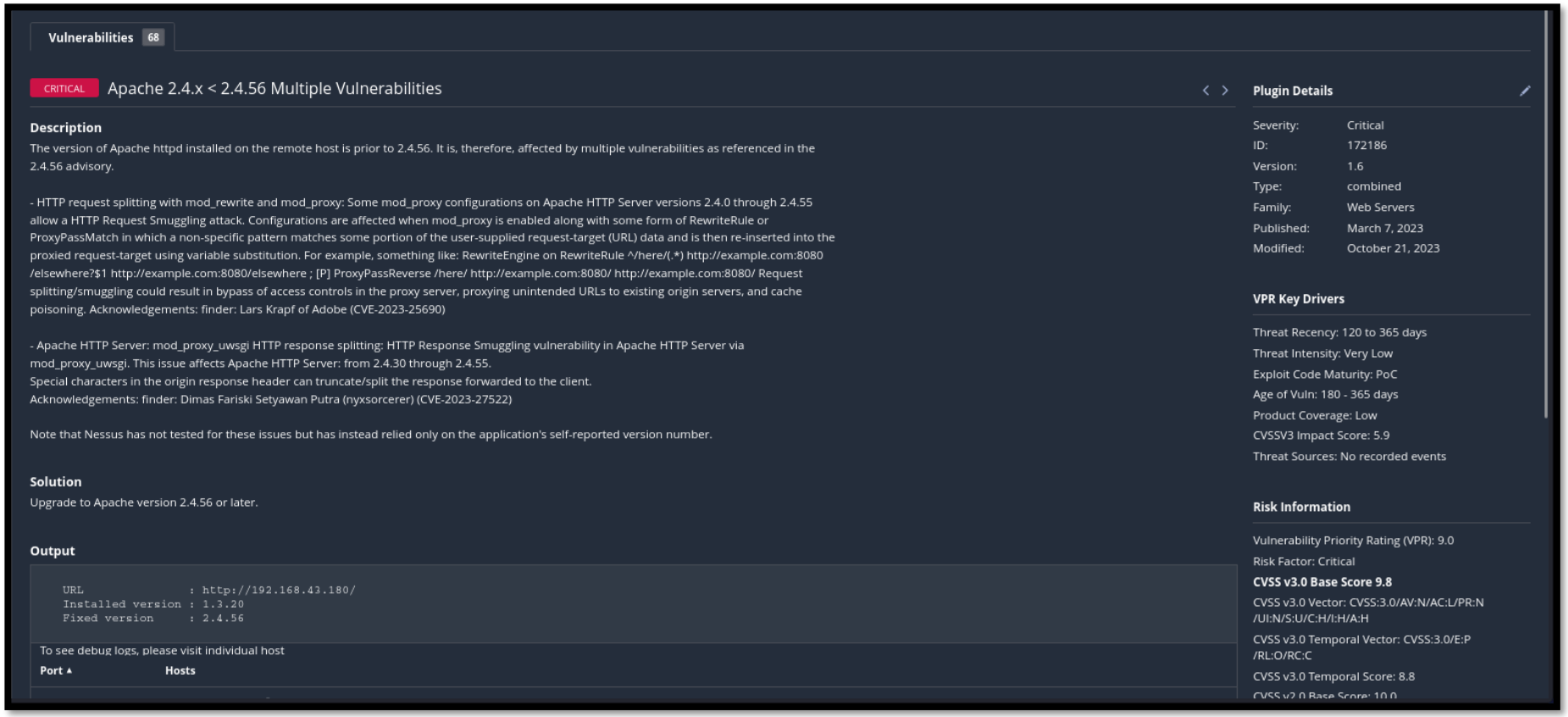
Required Resources:

- Personnel with expertise in Apache httpd administration and web server security.
- Access to non-production environments for testing.
- Backup systems and procedures.
- Change management processes for scheduling and deploying changes to production systems.
    - See also : http://archive.apache.org/dist/httpd/Announcement1.3.html

Changes to Network Configurations:

- In most cases, upgrading Apache httpd should not require significant changes to network configurations.
- However, ensure that firewalls, load balancers, and other network devices are configured to allow traffic to the new Apache httpd version on the appropriate ports.
- Update any network documentation to reflect the changes made during the upgrade process.

Apache 2.4.x < 2.4.56 Multiple Vulnerabilities



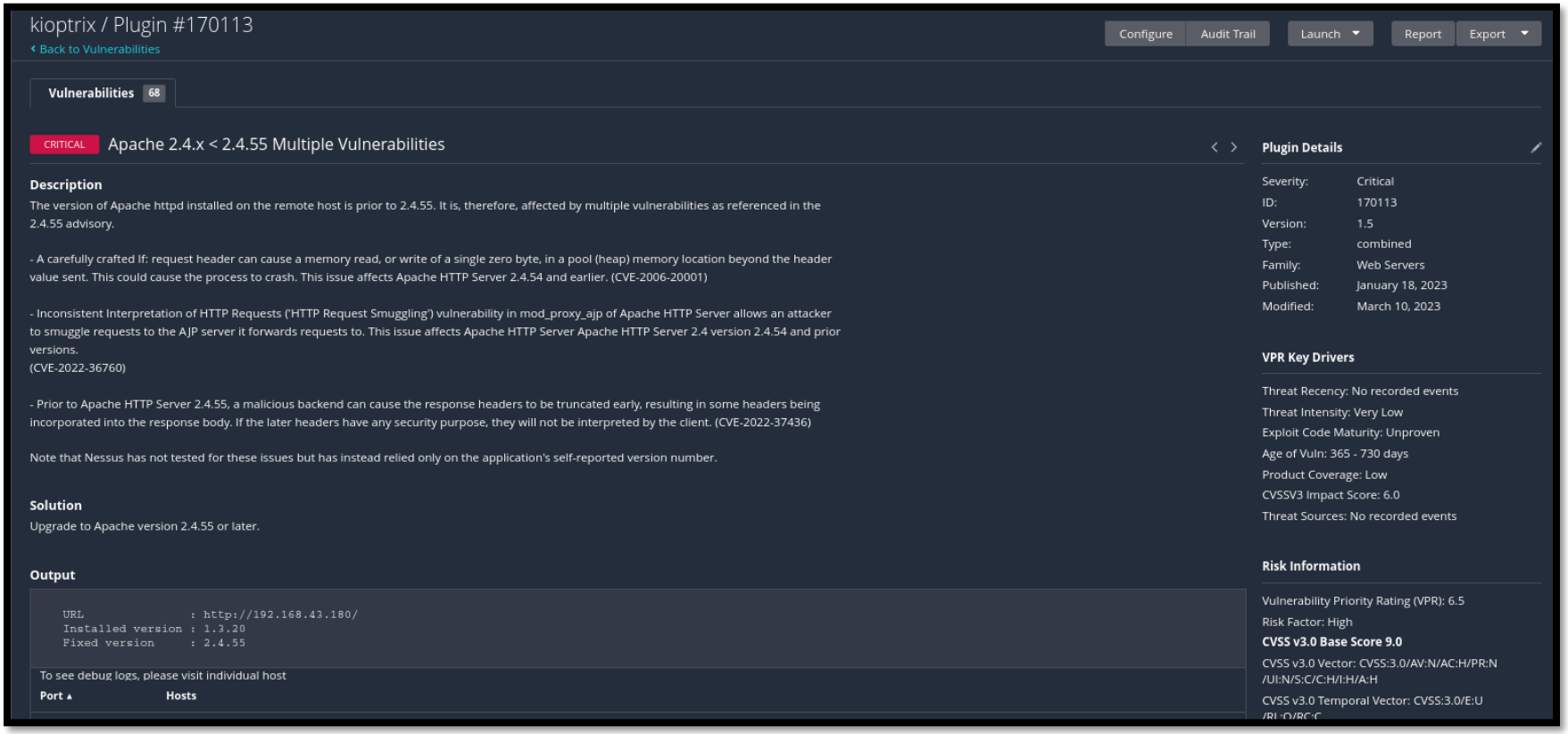
Mitigation Plan for Apache 2.4.x < 2.4.56:

1. **Assessment of Current Environment:**
    - Identify all systems running Apache web server versions 2.4.x < 2.4.56.

- Determine the criticality of these systems and the potential impact of the vulnerability.

2. **Patch Deployment:**
   - Obtain the latest Apache x.x.xx release from the official Apache website or distribution repository.
   - Schedule a maintenance window for applying the patch to affected systems.
   - Notify stakeholders and coordinate with relevant teams for the patching process.
3. **Remediation Steps:**
   - Stop the Apache web server service on each affected system.
   - Apply the patch by following the installation instructions provided with the update.
   - Restart the Apache service to ensure the changes take effect.
4. **Testing:**
   - Conduct thorough testing to ensure that the patch installation does not cause any adverse effects on the web server's functionality.
   - Test critical web applications and services hosted on the Apache servers to verify their continued operation post-patch.
5. **Timelines:**
   - Aim to complete the patch deployment within a defined maintenance window, considering the criticality of the vulnerability and the potential impact on operations.
   - Communicate the timeline to all stakeholders and adhere to any regulatory or compliance requirements.
6. **Required Resources:**
   - Allocate sufficient resources, including personnel and hardware, to execute the patching process efficiently.
   - Coordinate with IT support teams to address any technical challenges that may arise during the patch deployment.
7. **Network Configurations:**
   - Review and update firewall rules and access controls to ensure that the patched Apache servers are protected from unauthorized access.
   - Consider implementing additional security measures, such as intrusion detection/prevention systems, to enhance the overall security posture of the web server environment.

Apache 2.4.x < 2.4.55 Multiple Vulnerabilities



Mitigation Plan for Apache Apache 2.4.x < 2.4.55:

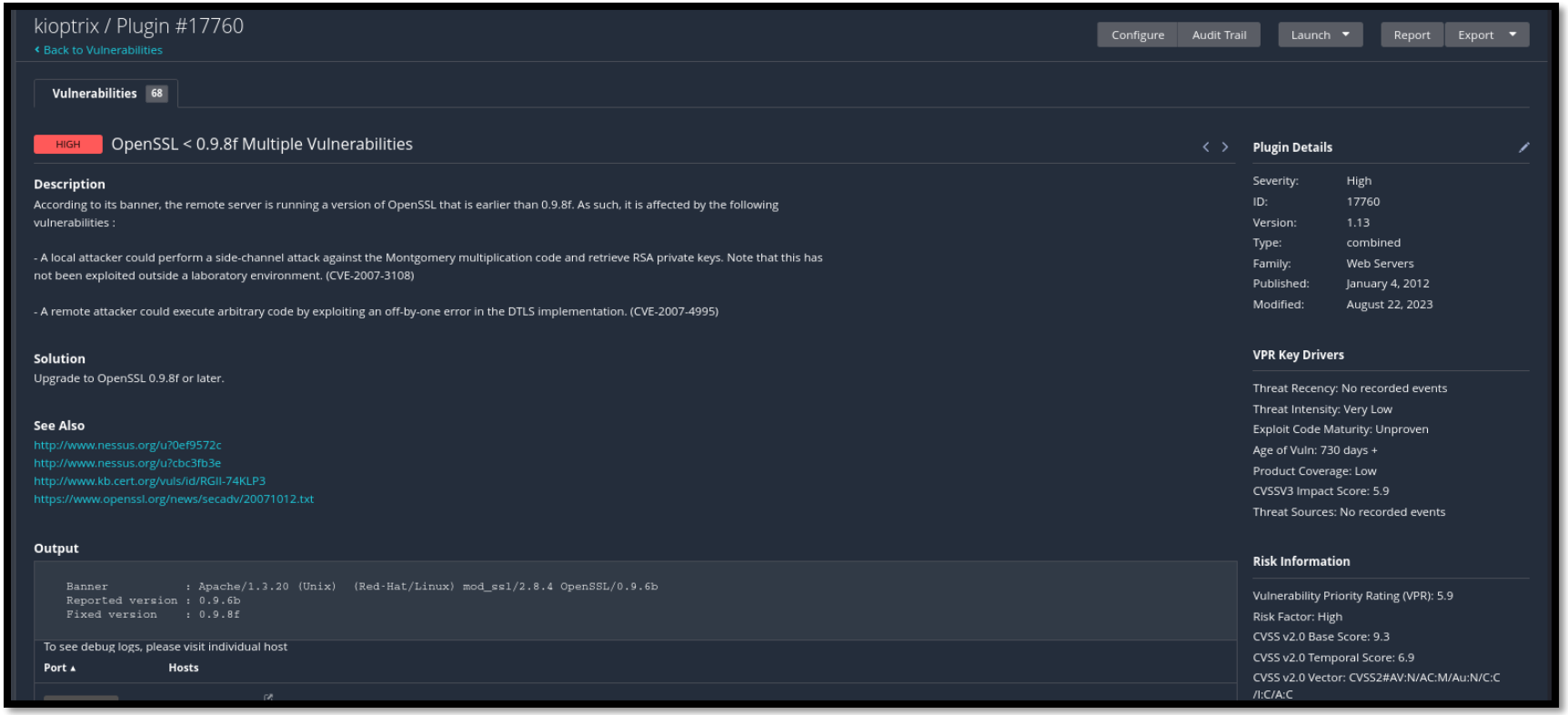
1. **Step-by-Step Instructions for Remediation**:
   a. **Update Apache**: The primary step is to update Apache to version x.x.xx or higher, which contains the necessary patches to address the vulnerabilities.

b. **Backup Configuration**: Before performing any updates, make sure to backup your existing Apache configuration files to avoid any loss of data or configuration settings.
c. **Download Latest Version**: Download the latest version of Apache from the official Apache website or through your distribution's package manager.

2. **Estimated Timelines**:
   The timeline for remediation will depend on the complexity of your Apache configuration and the availability of resources for updating and testing. However, the entire process, including updating, testing, and deploying the new version of Apache, can typically be completed within a few hours to a day.

3. **Required Resources or Changes to Network Configurations**:
   a. **Resource Allocation**: Allocate sufficient resources, including personnel and time, to perform the update and testing process.
   b. **Network Access**: Ensure that necessary network access is available to download the latest version of Apache and to test the updated installation.
   c. **Backup Resources**: Have backup resources available in case any issues arise during the update process, such as backups of configuration files and server data.

OpenSSL < 0.9.8f Multiple Vulnerabilities



Mitigation Plan for OpenSSL < 0.9.8f

Step 1: Identify Affected Systems

- Use vulnerability scanners or manual inspection to identify systems running OpenSSL versions prior to 0.9.8f.

Step 2: Prioritize Systems

- Prioritize systems based on their criticality to the organization and the sensitivity of the data they handle.

Step 3: Patching or Upgrading OpenSSL

- For systems running OpenSSL versions < 0.9.8f, patch or upgrade OpenSSL to version 0.9.8f or higher.
- Obtain the latest version of OpenSSL from the official website or package repositories.
- Ensure compatibility with existing applications and dependencies before applying the update.
- Test the patch or upgrade in a controlled environment to verify its effectiveness and compatibility.

Estimated Timeline:

- Immediate: Begin the process of patching or upgrading OpenSSL on critical systems.
- Within 1-2 weeks: Complete patching or upgrading OpenSSL on all identified systems.

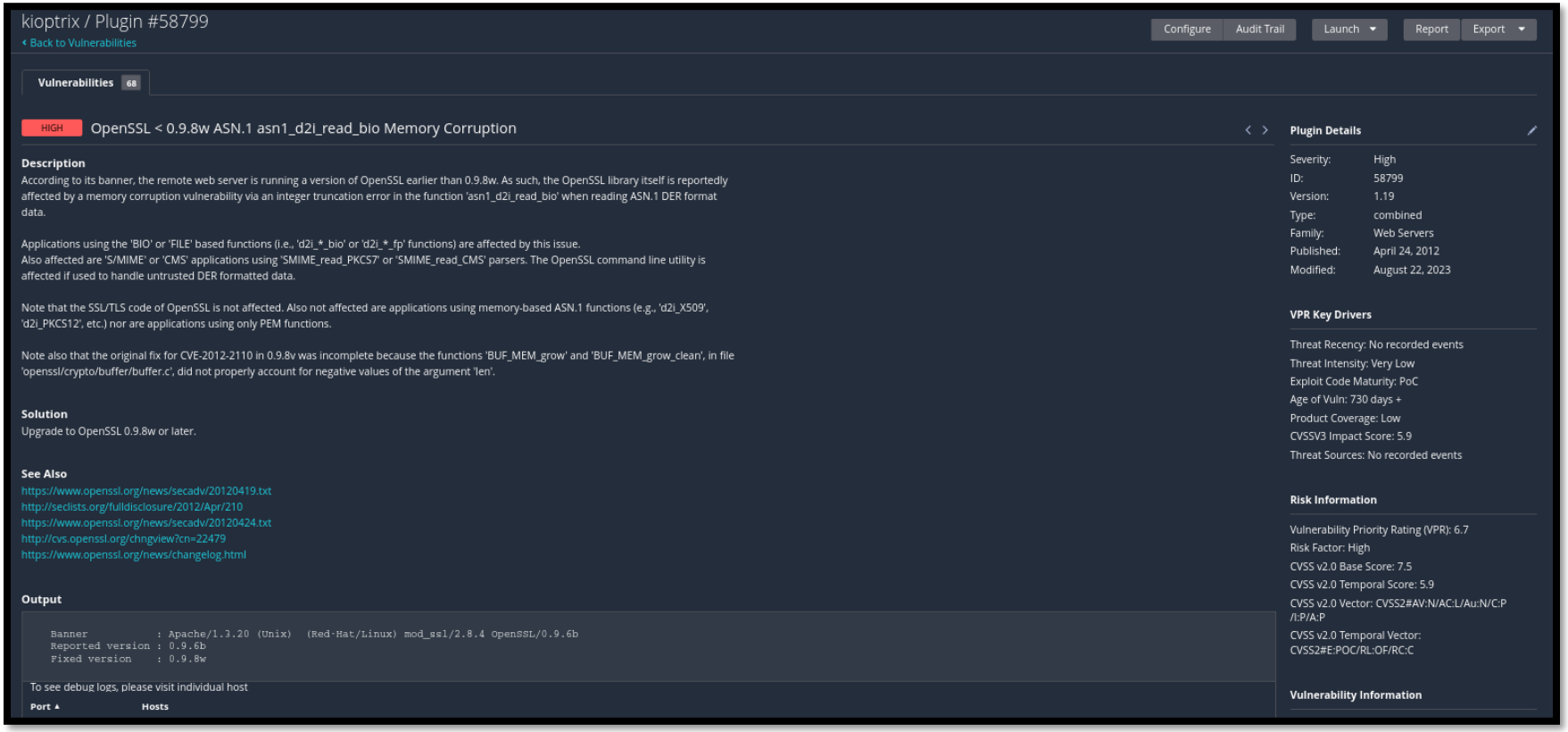
Step 4: Network Configuration Changes

- Update firewall rules or network access controls to restrict access to systems that have not been patched or upgraded.
- Implement additional network security measures to mitigate potential attacks targeting vulnerable systems.

Additional Resources

http://www.nessus.org/u?0ef9572c
http://www.nessus.org/u?cbc3fb3e
http://www.kb.cert.org/vuls/id/RGII-74KLP3
https://www.openssl.org/news/secadv/20071012.txt

OpenSSL < 0.9.8w ASN.1 asn1_d2i_read_bio Memory Corruption



Mitigation plan for OpenSSL < 0.9.8w ASN.1 asn1_d2i_read_bio Memory Corruption vulnerability

Patch OpenSSL:

- Obtain the latest version of OpenSSL (0.9.8w or higher) from the official website or your distribution's package manager.
- Apply the patch to all affected systems. This may involve downloading and compiling the source code or installing updated packages.

2. **Deploy the Patch**:
    - Schedule a maintenance window to apply the patch.
    - Notify all relevant stakeholders about the planned maintenance and potential downtime.
    - Deploy the patch to all affected systems, including servers, workstations, and any other devices running OpenSSL.
3. **Testing**:
    - Conduct thorough testing to ensure that the patch does not introduce any new issues or vulnerabilities.
    - Test the functionality of applications and services that rely on OpenSSL to ensure they continue to work as expected.
4. **Monitor for Anomalies**:
    - Implement monitoring solutions to detect any abnormal behavior or potential security incidents related to OpenSSL.
    - Monitor system logs, network traffic, and other relevant indicators for signs of exploitation or compromise.
5. **Timeline**:
    - The timeline for remediation will depend on the size and complexity of your environment, as well as any regulatory requirements or organizational policies.
    - Aim to apply the patch as soon as possible after it becomes available to minimize the window of exposure to the vulnerability.

Additional Resources

https://www.openssl.org/news/secadv/20120419.txt

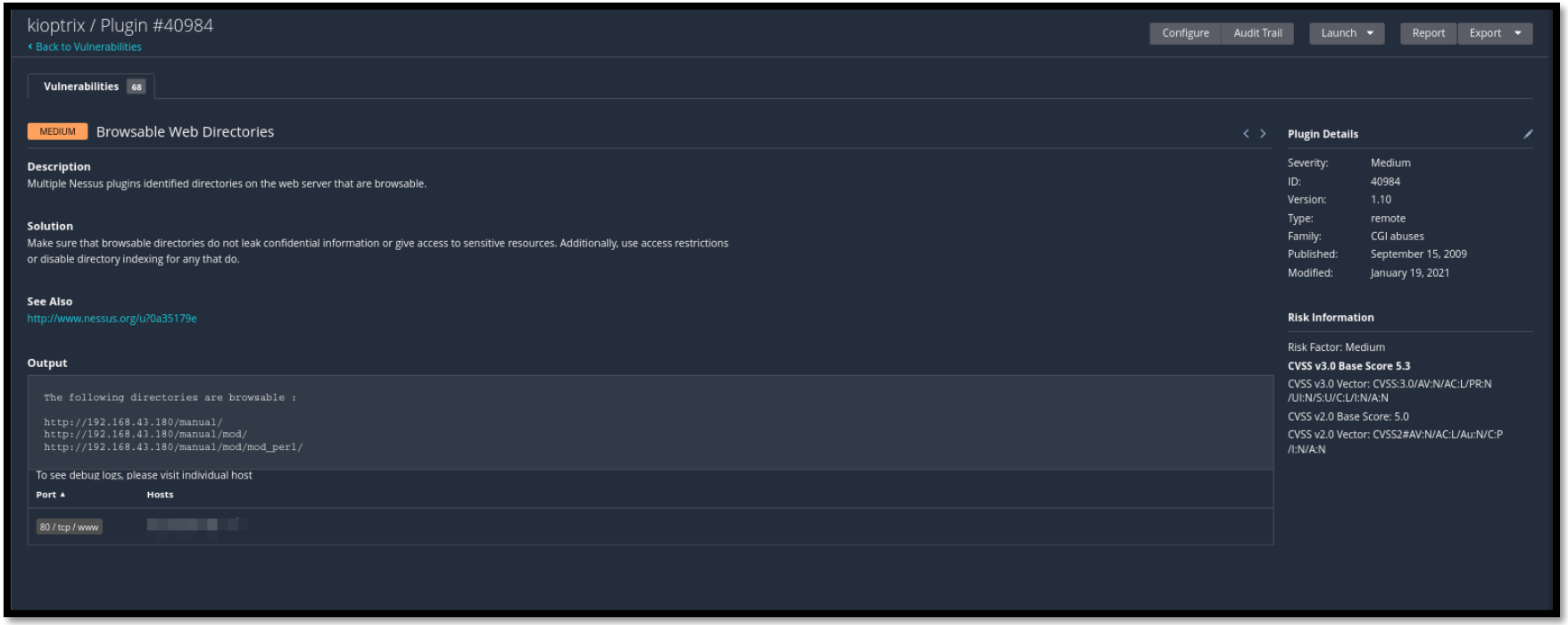
http://seclists.org/fulldisclosure/2012/Apr/210

https://www.openssl.org/news/secadv/20120424.txt

http://cvs.openssl.org/chngview?cn=22479

https://www.openssl.org/news/changelog.html

Browsable Web Directories



Mitigation Plan for Browsable Web Directories:

1. Identify all publicly accessible web directories on your network: Use web scanning tools or automated vulnerability scanners to identify all directories that are browsable.
2. Disable directory browsing: Configure your web server (e.g., Apache, Nginx) to disable directory browsing. This can typically be done by modifying the server configuration files (e.g., httpd.conf for Apache) and setting the "Options" directive to disable "Indexes" or "AutoIndex".
3. Implement access controls: Restrict access to sensitive directories by using access control mechanisms such as password protection, IP whitelisting, or user authentication. This will help prevent unauthorized access to directory listings.

4. Remove unnecessary directories: Review the content of each directory and remove any unnecessary or unused directories. This will help reduce the attack surface and minimize the risk of exposure.
5. Regularly monitor for new directories: Implement regular scans or automated monitoring to detect any new directories that may become browsable inadvertently. This will help ensure ongoing compliance with security best practices.

Estimated Timelines:

- Identification of publicly accessible web directories: 1-2 days
- Disabling directory browsing and implementing access controls: 1-2 days
- Removal of unnecessary directories: Variable, depending on the number and complexity of directories
- Regular monitoring: Ongoing, with periodic checks scheduled at regular intervals (e.g., weekly, monthly)

Additional Recommendations

- Regularly update Apache HTTP Server and other software components to the latest stable versions to mitigate the risk of known vulnerabilities.

- Implement a robust firewall configuration to restrict access to Apache servers and minimize the attack surface.
- Employ intrusion detection and prevention systems (IDPS) to detect and block malicious activities targeting Apache servers.

- Stay informed about security advisories and patches released by the Apache Software Foundation and other relevant sources to proactively address emerging threats.
-
- Consider implementing intrusion detection and prevention systems to detect and block potential attacks targeting OpenSSL vulnerabilities.

- Educate system administrators and users about the importance of keeping software and systems up to date to mitigate security risks.

- Stay informed about security updates and patches released by OpenSSL and other software vendors to proactively address vulnerabilities.

- Educate website administrators and developers about the importance of securing web directories and following best practices for web server configuration.

- Consider implementing a web application firewall (WAF) to provide an additional layer of defense against malicious attacks targeting web directories.