Digital Forensic Investigation and Network Traffic Analysis Report

This report details a digital forensic investigation conducted on a packet capture (PCAP) file from a suspected cyberattack. The analysis focused on identifying malicious activities, extracting forensic evidence, and pinpointing security vulnerabilities within the network. Key findings includes the exploitation of an unsecured FTP server allowing anonymous access, the upload of malicious tools such as Pwdump7.exe, and potential covert communication attempts via HTTP. The investigation highlights critical misconfigurations and provides actionable recommendations to mitigate future risks.

Objectives

The investigation aimed to:

- ❖ Analyze captured network traffic for evidence of malicious activity.
- ❖ Extract digital forensic artifacts, including IP addresses, credentials, and transferred files.
- ❖ Identify security misconfigurations contributing to the breach.
- ❖ Answer specific forensic questions based on observed network behavior.

Tools Utilized

The following tools were employed for the analysis:

- Wireshark: For packet-level inspection, protocol stream analysis, and object extraction.
- NetworkMiner: For session reconstruction and recovery of transferred files from the PCAP.
- Steghide: Prepared for detecting hidden messages in images (not used in this phase).
- John the Ripper/Hashcat: Available for cracking password hashes, though no hashes were extracted in this analysis.

Investigation Methodology

The investigation followed a systematic approach to analyze the PCAP file and extract actionable forensic evidence:

- Initial PCAP Analysis

The PCAP file was loaded into Wireshark, and the protocol hierarchy was examined, revealing significant FTP and HTTP activity. These protocols were prioritized for deeper inspection due to their common use in cyberattacks.

- FTP Traffic Analysis

Using Wireshark's display filter ftp.request.command, FTP command and response exchanges were analyzed to identify login behaviors and authentication patterns. The filter ftp.request.command == "STOR" was applied to focus on file upload activities, and the "Follow TCP Stream" feature was used to reconstruct session details.

- Endpoint Identification

Traffic flow analysis identified the destination IP address consistently receiving FTP commands and uploads, marking it as the target of the attack.

- Service Banner Extraction

The FTP welcome message was retrieved, identifying the target server as running Microsoft FTP Service on Windows XP, an outdated and vulnerable platform. The attacker leveraged anonymous login, which required no authentication credentials.

- File Transfer Analysis

Examination of FTP uploads revealed the transfer of malicious files, including Pwdump7.exe (a tool for extracting Windows password hashes), libeay32.dll, and hashlist.txt. These files suggest post-exploitation activities aimed at credential harvesting.

- HTTP Traffic Inspection

HTTP traffic was filtered using http.response.code == 404 to investigate potential covert communication via error pages. While no steganographic content was confirmed, these responses were flagged for further analysis to rule out hidden data transmission.

Key Observations

- Exploitation of FTP Server: The attacker exploited a misconfigured FTP server that allowed anonymous access without requiring authentication. All FTP communications were unencrypted, exposing usernames, passwords, and file contents to interception.
- Vulnerable System: The FTP server's service banner indicated it was running on Windows XP, a deprecated operating system with known vulnerabilities.
- Malicious File Uploads: The attacker uploaded Pwdump7.exe, a tool commonly used for extracting Windows password hashes, along with supporting files (libeay32.dll and hashlist.txt), indicating preparation for credential dumping.

- ❖ Potential Covert Communication: HTTP 404 responses suggest possible attempts to hide resources or transmit data covertly. While no steganographic content was identified, further analysis is warranted.
- ❖ Security Gaps: The environment lacked fundamental security controls, including access restrictions, encryption, and intrusion detection mechanisms, making it highly susceptible to exploitation.

Forensic Questions and Answers

- What was the IP address of the target machine?

Answer: 192.168.158.131

- What service banner was presented during the FTP connection?

Answer: 220-Microsoft FTP Service (running on Windows XP)

- What was the name of the executable file uploaded by the attacker using the FTP service?

Answer: Pwdump7.exe

- What was the administrator password?

Answer: No administrator password was used. The attacker logged in using:

  - Username: anonymous
  - Password: anonymous

- What was the website's hidden "Secret" message?

Answer: No hidden message was confirmed in this phase. HTTP 404 responses suggest potential covert communication, but further steganographic analysis is required.

Conclusions

The investigation successfully identified a cybersecurity breach facilitated by an unsecured FTP server. The attacker exploited anonymous login functionality to upload malicious tools, likely preparing for credential harvesting. The use of an outdated Windows XP system and unencrypted FTP communications significantly increased the attack surface. While HTTP analysis hinted at covert communication, no hidden messages were confirmed in this phase.

The findings underscore the critical need for robust security configurations to prevent similar incidents. Key vulnerabilities included:

- Lack of authentication requirements for FTP access.

- Use of deprecated and unpatched systems (Windows XP).
- Absence of encryption for sensitive data transfers.
- Insufficient monitoring and detection mechanisms.

Recommendations

To strengthen network security and prevent future breaches, the following measures are recommended:

- Implement Access Controls: Enforce strong authentication mechanisms for FTP and other services, disabling anonymous access and requiring complex credentials.
- Upgrade Legacy Systems: Replace Windows XP with a supported operating system, ensuring regular security patches and updates.
- Enable Encryption: Use secure protocols (e.g., FTPS or SFTP) to encrypt data in transit, protecting sensitive information from interception.
- Conduct Regular Audits: Perform periodic security assessments to identify and remediate misconfigurations and vulnerabilities.
- Enhance Monitoring: Deploy intrusion detection and prevention systems (IDPS) to detect and respond to suspicious activities in real time.
- Adopt Least-Privilege Policies: Restrict file upload capabilities to authorized users only, minimizing the risk of malicious uploads.
- Employee Training: Educate staff on secure configuration practices and the risks of outdated systems and protocols.