Security incident report

Section 1: Identify the network protocol involved in the incident

The network protocol involved in the incident is the exploitation of Hypertext transfer protocol (HTTP). The machine accessed `yummyrecipesforme.com` and received the correct IP address. After a successful connection, the site responded with the HTTP GET request. The website contained malicious code that redirected the user to `greatrecipesforme.com` after prompting for a file download. The machine resolved the IP for `greatrecipesforme.com` and initiated an HTTP connection, confirming the malicious redirection and subsequent access to the fake website.

The attack was made possible due to the initial compromise of the website, where malicious code was injected, leading to the redirection and unauthorized download of malware.

Section 2: Document the incident

Summary of the Incident

A former employee, referred to as "the baker," conducted a brute force attack to gain unauthorized access to the admin panel of the website, yummyrecipesforme.com. They guessed the administrative account's password, which was left at its default setting, and then modified the website's source code. This alteration included embedding a JavaScript function that tricked visitors into downloading a malware-infected file. The malware redirected users to a fake website, greatrecipesforme.com. Several customers reported experiencing issues after interacting with the infected site, leading to the discovery of the breach.

Identified Problems

1. Weak Password Policy : The administrative account's password was left at its default setting, making it vulnerable to brute force attacks.
2. Lack of Brute Force Protection : No security measures were in place to detect or block multiple failed login attempts.
3. Inadequate Monitoring and Alerts : There were no alerts for unauthorized access

or changes to the website's source code.
4. No File Integrity Monitoring : The alteration of the website's source code was not detected immediately, indicating a lack of integrity checks.

Testing Activities Involved

1. Sandbox Environment Setup : A secure environment was created to replicate and observe the suspicious behavior of the website.
2. Network Traffic Analysis : Tools like `tcpdump` were used to capture and analyze network traffic, documenting the behavior of the browser when interacting with the compromised website.
3. Behavioral Analysis of Downloaded File : The downloaded executable file was executed in the sandbox to observe its behavior, which included redirecting the browser to a fake website.
4. Source Code Review : A senior analyst reviewed the website's source code and identified the malicious JavaScript that prompted visitors to download the malware.

Analysis Work Involved from the Log File

1. DNS Requests Analysis : The log shows that the user's machine made DNS requests to resolve the IP addresses for both `yummyrecipesforme.com` and `greatrecipesforme.com`. The DNS server responded with the IP addresses `203.0.113.22` for `yummyrecipesforme.com` and `192.0.2.17` for `greatrecipesforme.com`. This indicates that the machine was redirected from the legitimate website to the malicious site.

2. HTTP Requests Analysis : The user's machine initiated HTTP connections to both `yummyrecipesforme.com` and `greatrecipesforme.com` on port 80. The logs show typical HTTP traffic, including TCP handshake sequences (SYN, SYN-ACK, ACK) and HTTP GET requests.
   - The HTTP request sequence started with `yummyrecipesforme.com`, which served the initial webpage. Subsequently, there was an HTTP GET request to `greatrecipesforme.com`, indicating that the malware had redirected the browser to this site.

3. HTTP Download and Redirection : The HTTP traffic for `yummyrecipesforme.com` likely included the malicious JavaScript code that prompted the download of an executable file. The presence of significant traffic on port 80 confirms that content was being served and potentially downloaded.

4. Redirection Confirmation : After visiting `yummyrecipesforme.com`, the browser made a DNS request and then HTTP GET request to `greatrecipesforme.com`, confirming the redirection caused by the embedded JavaScript or executed malware.

Conclusion

The analysis of the log file confirms the sequence of the attack:

- Initial DNS Resolution and Access : The machine accessed `yummyrecipesforme.com` and received the correct IP address. After a successful connection, the site responded with the HTTP GET request.
- Malicious Activity Triggered : The website contained malicious code that redirected the user to `greatrecipesforme.com` after prompting for a file download.
- Redirection and Malicious Site Access : The machine resolved the IP for `greatrecipesforme.com` and initiated an HTTP connection, confirming the malicious redirection and subsequent access to the fake website.

The attack was made possible due to the initial compromise of the website, where malicious code was injected, leading to the redirection and unauthorized download of malware.

Conclusion About the Root Cause of the Attack

The root cause of the attack was the use of a weak, default administrative password, which allowed the attacker to perform a brute force attack and gain access to the web host. The lack of brute force protection and monitoring controls facilitated unauthorized access and modifications to the website's source code, leading to the deployment of malware that redirected users to a fake, malicious website.

Section 3: Recommend one remediation for brute force attacks

To protect against brute force attacks and prevent unauthorized access in the admin panel, mitigating malware injection in source code, it's essential to implement a range of security measures. Here are some best practices:

Preventing Brute Force Attacks

1. Use Strong Password Policies :
   - Enforce complex passwords with a mix of upper and lower case letters, numbers, and special characters.

- Implement multi-factor authentication (MFA) to add an extra layer of security.
- Use biometric authentication (fingerprint, facial recognition) if supported.

2. Account Lockout Mechanism :

- Set up an account lockout policy that temporarily disables an account after a certain number of failed login attempts.
- Implement CAPTCHA or reCAPTCHA to distinguish between human and automated login attempts.

3. Rate Limiting :

- Apply rate limiting on login attempts to slow down or block repeated access attempts from the same IP address.

4. Monitoring and Alerts :

- Implement logging and monitoring for login attempts and other critical actions.
- Set up alerts for unusual login patterns or failed attempts.

5. Use of Progressive Delays :

- Implement increasing time delays between each failed login attempt to slow down brute force attacks.

6. Secure API Endpoints :

- Protect API endpoints with strong authentication and rate limiting mechanisms.
- Use token-based authentication and rotate tokens periodically.

Preventing Unauthorized Access in Admin panel – Using Advanced Firewall Configurations and Best Practices

to protect the internal network from external attacks of disgruntled former employees

1. IP Whitelisting :

- Allow only specific, trusted IP addresses or IP ranges to access critical parts of your network. This ensures that only approved external IPs can connect to the internal network.

2. IP Blacklisting :

- Block known IP addresses associated with former employees or suspicious activities. Use this as a preventive measure to deny access.

3. Geo-blocking :
   - Restrict access to your network based on geographic locations. If your employees are only supposed to connect from certain regions, block traffic from countries or regions where former employees reside.

4. Access Control Lists (ACLs) :
   - Implement strict ACLs on the firewall to control which devices or users can access specific network segments. Use these rules to block external access from unauthorized IP addresses.

5. VPN Access Restrictions :
   - Ensure that VPN access is disabled for former employees. Implement multi-factor authentication (MFA) for VPN logins and use short-lived VPN session tokens.
   - Restrict VPN access by IP address and use client certificates for an additional layer of security.
   - Ensure that the admin panel is only accessible over HTTPS to encrypt data in transit.

6. Zero Trust Network Architecture :
   - Adopt a Zero Trust model where no user or device is trusted by default, even if they are within the internal network. All access requests are verified and authenticated based on strict identity and access management policies.

7. Network Segmentation :
   - Segment your network to isolate critical systems from general user access. Apply strict firewall rules between these segments to control and monitor traffic.

8. Intrusion Detection and Prevention Systems (IDPS) :
   - Deploy IDPS to monitor network traffic and detect suspicious activities or attempted breaches. Set up alerts for unusual behavior, such as failed login attempts or access from unexpected IP addresses.

9. Use of Deception Technology :
   - Deploy honeypots or decoy systems to attract and detect unauthorized access attempts. This helps in identifying former employees or malicious actors trying to gain access.

10. Monitoring and Logging :
    - Continuously monitor and log all access attempts to your network. Set up alerts for access attempts from blocked IPs or unusual patterns indicative of

unauthorized access attempts.

11. Restrict Remote Access :
    - Limit remote access to only those who need it. Use strong authentication mechanisms like certificates and MFA for all remote connections.
    - Implement session timeouts and limit the duration of remote access sessions.

Additional Recommendations

- Disable or Remove Accounts and Unused features : Ensure that all accounts associated with former employees are completely disabled or removed from your systems, including third-party services, to prevent unauthorized access.

- Regular Audits : Regularly audit firewall rules, user accounts, and network access permissions to ensure they align with current employee roles and security policies.

- Update Threat Intelligence : Use updated threat intelligence feeds to keep track of known malicious IP addresses and block them at the firewall level.

By applying these configurations and practices, you can significantly reduce the risk of unauthorized access to your network by former employees or other external threats.

Preventing Malware Injection in Source Code

1. Use Source Control :
    - Use a version control system like Git to track changes and monitor for unauthorized modifications to the codebase.
    - Implement access controls and require code reviews for all changes.

2. Implement Code Reviews :
    - Conduct thorough code reviews to identify and remove potential vulnerabilities.
    - Use automated tools to scan for common security issues, such as malicious code patterns or unsafe coding practices.

3. Secure Development Environment :
    - Ensure that development environments are secure and isolated from

production environments.
   - Restrict access to the source code and development environments to authorized personnel only.

4. Use Security Tools :
   - Employ Static Application Security Testing (SAST) and Dynamic Application Security Testing (DAST) tools to identify vulnerabilities in your code.
   - Use tools like `git-secrets` to prevent committing sensitive data or credentials to the repository.

5. Third-Party Libraries :
   - Regularly update third-party libraries and frameworks to protect against known vulnerabilities.
   - Use tools like `OWASP Dependency-Check` to identify and manage vulnerabilities in dependencies.

6. Continuous Integration/Continuous Deployment (CI/CD) Security :
   - Integrate security checks into the CI/CD pipeline to detect and prevent malicious code from being deployed.
   - Use automated tests to verify code integrity and functionality.

7. Training and Awareness :
   - Educate developers on secure coding practices and the risks associated with malware injection.
   - Conduct regular security awareness training for all team members.