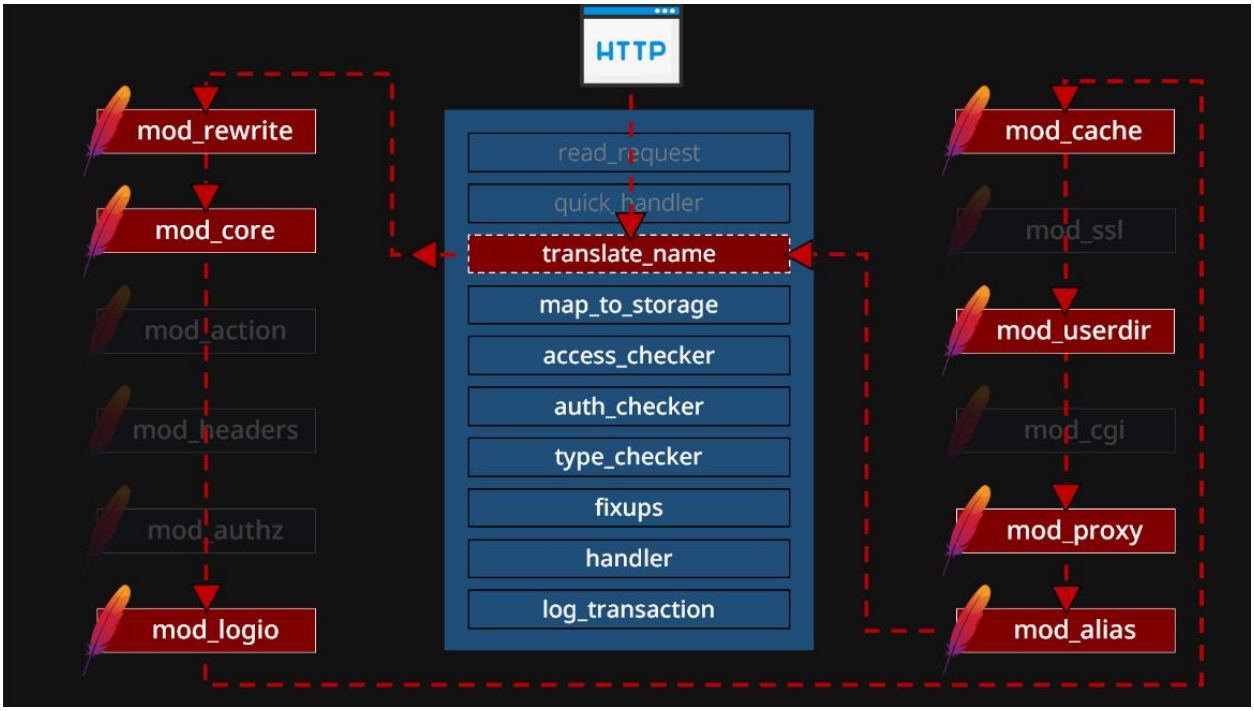
# Serianu Advisory on Confusion Attacks in Apache HTTP Server




## Threat Scenario: Attacks on Apache HTTP Server

Architectural vulnerabilities within HTTP Server (a widely used web server software) allow attackers to gain root access remotely.

Apache HTTP server operates through a modular design, where hundreds of small modules work together to handle HTTP requests. A HTTP request passes through various phases which are modified by the modules as needed. Each module focuses on its specific task; however, complexity increases when scaled to hundreds of modules. The modules' lack of deep understanding and the absence of stringent development guidelines create gaps and inconsistencies that make the system vulnerable to potential exploitation.

## Threat Targets:

- **Financial Services** - online banking platforms that interpret user inputs or session data
- **E-Commerce** - online stores and trading platforms using payment processing applications
- **Healthcare** – Hospitals handling patient management systems or online appointment scheduling
- **Government** - misconfigurations in the servers handling public services and websites that use Apache HTTP Server are at risk.
- **Education** – Online learning platforms, administrative portals and websites handling student records or online grading systems could be vulnerable if the web servers are using Apache HTTP Server.
- **Technology and Software Companies** – Tech companies providing software services or hosting solutions are at risk if they rely on Apache HTTP Server for their web infrastructure.

## Threat Actors:

- **Insider Threat Actors** – Disgruntled employees, malicious insiders, current or former employees with knowledge of the systems might exploit vulnerabilities to cause harm or steal data.
- **Automatic attack tools** – Scanners or botnets deployed by attackers at large scale to automatically detect and exploit this vulnerability.
- **Cybercriminals / Organized Crime Groups** – Individuals or organized groups may exploit this vulnerability for financial gain by stealing credit card information, personal data or login credentials.

## Threat Indicators:

- **Server Configuration Changes** – Unauthorized configuration changes to server configuration files like `**httpd.conf**` or `.**htaccess**`
- **Unusual Server Behavior** – Unexpected Errors like `500 Internal Server Error` which could indicate issues in request handling or misinterpretation.
- **Abnormal Traffic Patterns** – High volume of requests from a single Source IP address which could indicate automated attempts to exploit the vulnerability
- **Suspicious HTTP Request Methods** – Unusual use of HTTP methods like PUT or DELETE that are not used by regular users.
- **Alerts from Web Application Firewalls (WAFs) and Intrusion Detection Systems (IDS)** – warnings or blocks triggered by security tools which might indicate suspicious activities or exploit attempts.
- **Logs and Anomalies** – Unexpected Log entries showing access to or manipulation of sensitive resources, unexpected HTTP status codes or unusual user-agent strings.
- **Presence of Exploitation Tools** – Presence of tools or scripts commonly used for exploiting Webservers

## Risk Scenario(s):

- **Data Breaches** – Attackers may gain unauthorized access to sensitive data including user credentials, financial information or business data.
- **Website Defacement** – Attackers may modify or deface web pages damaging reputation and trustworthiness of the organization.
- **Data Manipulation** – Attackers could alter data stored on the server such as user profiles, financial records and other user data stored on the Apache server.
- **Financial Loss** – Fraudulent Transactions; exploiting vulnerabilities in payment systems could lead to financial fraud, theft or unauthorized transactions.
- **Legal and Compliance Costs** – Data breaches or service disruptions could lead to legal actions, regulatory fines and compliance costs.
- **Spread of Malware** – Attackers may use compromised servers to distribute malware across the network as well as deploy chain attacks where they leverage other vulnerabilities existing within the network.

- **Intellectual Property Theft** – Stolen Trade Secrets where attackers may gain access to proprietary algorithms, designs or other intellectual property impacting competitive advantage and innovation.
- **Exploitation of Trust/Third-party Relationships** – Attacks on Partner Systems where if the vulnerable server interacts with other systems or partners, attackers can exploit the trust relationship to target other connected devices/systems.
- **Denial of Service (DoS)** – Confusion attacks can overwhelm the server leading to service disruptions or outages affecting availability for legitimate users.

**Risk Exposure:** Loss of Funds (Fraud), Information Disclosure, Denial of Service, Remote Code Execution (RCE), Unauthorized Access, Data Corruption, Cache Poisoning

## Recommended Threat Mitigation Actions:

- **Update and Patch Regularly** - Keeping the Apache HTTP Server updated with the latest security patches. Regularly check for and apply security patches provided by the Apache Software Foundation.
- **Configuring the webserver securely** – Harden web server configurations and review configuration files like `**httpd.conf**` and `.**htaccess**` to ensure they follow security best practices. Disable unnecessary modules and services.
- **Set Proper Permissions** – Restrict file and directory permissions to the minimum required for operations.
- **Implement Access Controls** – Restricting access to sensitive areas using access control directives like `**Require**`, `**Allow**` or `**Deny**` to limit access to administrative interfaces and other critical resources.
- **Enable Security Modules** – Enable and configure Apache Security modules like `**mod_security**` for web application firewall functionality and `**mod_evasive**` to mitigate DoS attacks.
- **Implement HTTP Security Headers** – Configure security headers like `**X-Content-Type-Options**`, `**X-Frame-Options**` and `**Content-Security-Policy**` to enhance protection against certain types of attacks.
- **Monitoring Log Activity** – Monitor for suspicious activity around access and error logs
- **Regular Security Reviews** – Perform Vulnerability Scanning regularly to scan your server for vulnerabilities using scanning tools such as Nessus or manual testing. Engaging in Penetration Tests as well, to identify and address security weaknesses to ensure it is resilient to such attacks.
- **Deploy Web Application Firewalls (WAFs)** – to filter and monitor HTTP requests and responses which could help protect against various types of web-based attacks.
- **Implement Network Security Measures** – Applying Network segmentation to isolate Apache Servers from other critical systems and deployment of network firewalls to restrict access to the Apache server from unauthorized IP addresses or networks.
- **Educate and Train Personnel** – Train Administrators and developers to ensure they are aware of security best practices and understand the implications of misconfigurations.

Ensuring these parties stay informed about emerging security threats and best practices for securing Apache HTTP Servers and other webservers.

## Conclusion

It is critical you perform an analysis of your environment, validate that these controls have been implemented and that you have visibility on all transactions, vendor activities and changes being made to webservers and critical systems by your internal team. This will require the collective cooperation of IT, Risk and Audit.

We encourage recipients who are unsure of their security posture, unsure of their technical capabilities to implement the above recommendations and/or identify malicious activity or use of tools or techniques that seem malicious to contact us on the following:

**Helpdesk +254(0)716137017, Cybercrime hotline +254 771949475, email: Info@serianu.com.**