Bypassing F5 WAF for Data Exfiltration during Post-Exploitation

Avtandili Bichnigauri, Luka Shonia, Ilia Shonia

Georgian Technical University

bichnigauri_av@gtu.ge, shonia@gtu.ge, iliashoniafr@gmail.com

Abstract

The post-exploitation phase of a cyberattack is where adversaries achieve their ultimate objectives, often centered on the exfiltration of sensitive data. While Web Application Firewalls (WAFs), such as those from F5 Networks, serve as a critical defense layer, their reliance on static, signature-based detection can present a significant vulnerability. This article explores a practical methodology for bypassing F5 WAF protections to enable data exfiltration after an initial compromise. The technique involves a multi-layer encoding scheme (Base64, Reversal, and Hexadecimal encoding) applied to both the attacker's commands within the HTTP request and the stolen data within the HTTP response. This process effectively obfuscates malicious payloads from static rule sets. We detail the step-by-step methodology, discuss the inherent limitations of static analysis in WAFs, and present potential solutions for defenders, including dynamic analysis, parameter validation, and behavioral anomaly detection. The conclusion underscores the evolving threat landscape and proposes future research directions, such as employing custom encryption and asymmetric cryptography for more sophisticated evasion.

Keywords: Data Exfiltration, Post-Exploitation, F5 WAF, Web Application Firewall Bypass, Encoding, Obfuscation, Cybersecurity, Command and Control, C2, Static Analysis, Dynamic Analysis, SIEM, IDS/IPS.

1. Introduction.

In the modern cyber threat landscape, the initial compromise of a system is often merely the opening act. The true damage occurs during the **post-exploitation** phase, where attackers leverage their foothold to achieve malicious goals, primarily **data exfiltration**. Data exfiltration is the unauthorized transfer of sensitive information - such as intellectual property, financial records, or personal identifiable information (PII) from a victim's network to an attacker-controlled server. This can be executed through various vectors, including outbound emails, downloads to insecure devices, or direct uploads over network protocols.

To defend against such attacks, organizations deploy security controls like the F5 Web Application Firewall (WAF). The F5 WAF operates as a filter between a web application and the internet, inspecting HTTP/S traffic for known attack patterns, such as SQL injection (SQLi) commands, cross-site scripting (XSS) tags, or system commands. Its primary functionality is to block requests containing these malicious signatures in the URL, headers, or body.

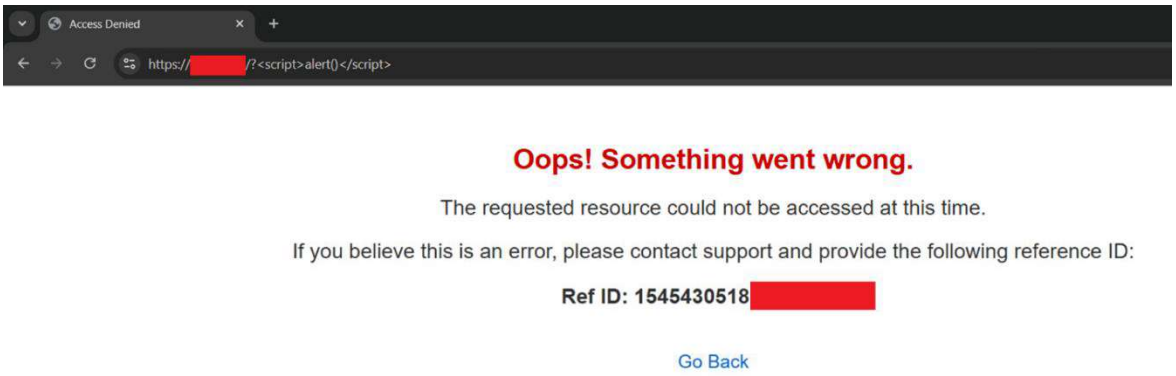
However, a significant challenge exists: many WAFs, including legacy or misconfigured F5 deployments, rely heavily on **static rule sets**. They check for predefined patterns but often lack the context to understand the intent behind an obfuscated payload. This paper presents a methodology demonstrating how an attacker, having already gained a foothold on a web server (e.g., through a vulnerable web application), can bypass F5 WAF's static rules to execute commands and exfiltrate data by strategically encoding the entire communication channel.

2.Methodology.

This methodology assumes the attacker has already achieved initial code execution on the target server, for instance, via a remote file inclusion (RFI) or a deserialization vulnerability. The goal is to interact with the underlying operating system to locate and exfiltrate sensitive data without triggering the F5 WAF. The core of the bypass technique lies in the dual-layer obfuscation of both the outgoing commands and the incoming data.

1. Bypassing the HTTP Request with Multi-Layer Encoding

An attacker cannot simply send a command like "**cat /etc/passwd**" in a POST request, as the WAF will easily flag and block it. Instead, the command must be disguised.



1. **Command Execution:** The attacker aims to run a system command, such as "**cat /etc/passwd**", to list files in a sensitive directory.
2. **Encoding Pipeline:** The command is passed through an encoding pipeline on the attacker's machine before being sent to the compromised server.

- **Step 1: Base64 Encode:** The command "**cat /etc/passwd**" is first Base64 encoded.
  - Result: "**Y2F0IC9ldGMvcGFzc3dk**"
- **Step 2: Reverse the String:** The Base64 string is then reversed.
  - Result: "**kd3czFGcvMGdl9CI0F2Y**" (Note: The reversal makes the string unrecognizable to Base64 decoders and pattern matchers).
- **Step 3: Convert to Hexadecimal:** The reversed string is finally converted to a hex string.
  - Result: "**6b6433637a464763764d47646c39434930463259**"

3. **Crafting the Malicious Request:** The attacker places this final hex payload into an HTTP parameter that the compromised server will process. For example:

The F5 WAF, performing a static check, sees only a benign-looking hex string. It does not match any known command signatures (ls, cat, /etc/passwd), so the request is allowed to pass through.

2. Processing and Decoding on the Compromised Server

The vulnerable application on the compromised server must be modified (or already possess the functionality) to process the **payload** parameter. A small piece of malicious code executes the following:

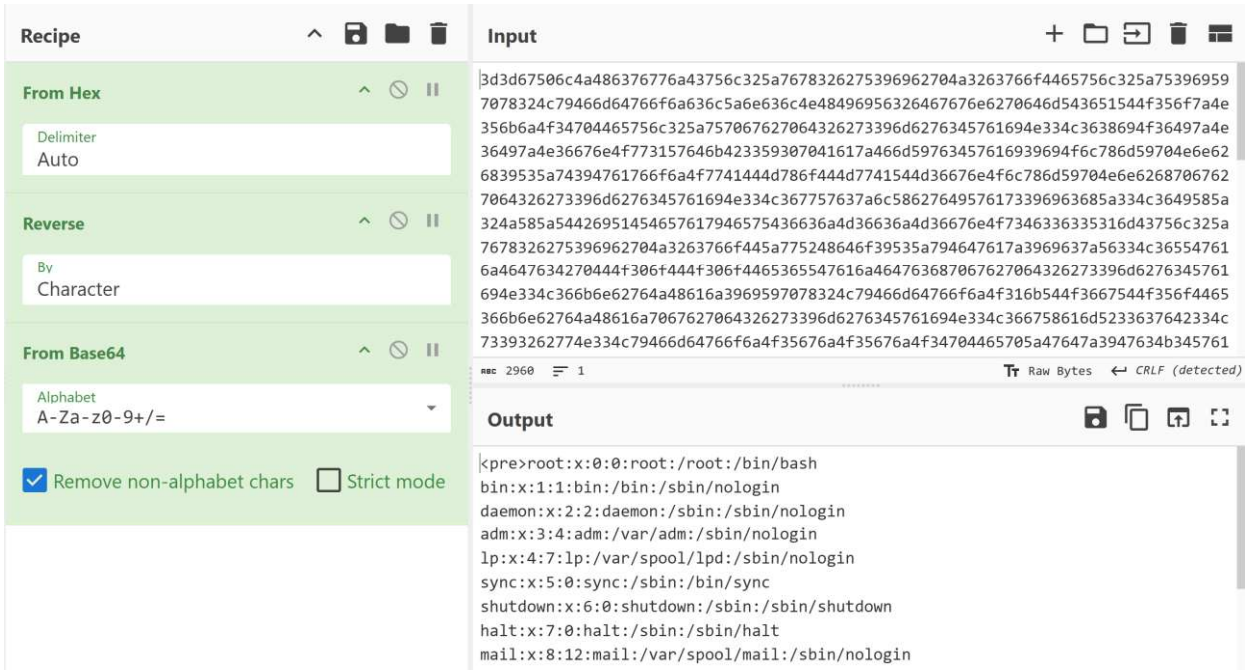
1. Receive the hex payload: "**6b6433637a464763764d47646c39434930463259**"
2. Decode it from hex back to a string: "**kd3czFGcvMGdl9CI0F2Y**"
3. Reverse the string to recover the Base64: "**Y2F0IC9ldGMvcGFzc3dk**"
4. Base64 decode it to recover the original command: "**cat /etc/passwd**"
5. Execute the command on the server's operating system.

3. Exfiltrating Data and Bypassing the HTTP Response

The WAF also often inspects outbound responses for signs of sensitive data (e.g., database dumps, password files). To bypass this, the same obfuscation principle is applied in reverse.

1. The server captures the output of the command "**cat /etc/passwd**".
2. Encoding Response Pipeline (Base64 Encode **->** Reverse **->** Hex).
3. The resulting hex string is placed into the HTTP response body and sent back to the attacker.
4. The F5 WAF sees a hex stream, not plaintext sensitive data, and allows the response to pass.

The attacker receives the hex payload, decodes it through the reverse pipeline (Hex **->** Reverse **->** Base64 Decode), and views the stolen data in clear text.



9.Discussion.

The success of this technique highlights a critical weakness in security controls that depend solely on static pattern matching. The F5 WAF, in this scenario, is effectively blind to the attack because the malicious intent is hidden behind a layer of encoding that is trivial for an attacker to implement.

This method is effective because:

- **It Obfuscates Signatures:** The static rules looking for "cat /etc/passwd" will not trigger on a hex-encoded and reversed string.
- **It Leverages Legitimate Functions:** Encoding like Base64 and Hex are widely used in legitimate web traffic, making it difficult to block them outright without causing false positives.
- **It Targets the Post-Exploitation Phase:** The attack assumes a compromised endpoint, shifting the focus from perimeter defense to the integrity of the application logic itself.

The limitations of this approach are primarily on the attacker's side: it requires a non-interactive, command-and-response cycle, which can be slower than an interactive shell. Furthermore, the need for a custom decoder on the server implies a persistent backdoor or the ability to inject code repeatedly.

From a defender's perspective, this underscores that a WAF is not a silver bullet. The proposed solutions must involve a defense-in-depth strategy:

- **Dynamic Analysis:** Next-generation WAFs should incorporate dynamic analysis, which involves decoding and de-obfuscating parameters in a sandboxed environment to inspect their true content before allowing the request to proceed.
- **Parameter Monitoring and Expectation of Values:** Applications should enforce strict input validation. If a parameter like **user_id** is expected to be an integer, any value containing Base64 or hex characters should be rejected outright.
- **Behavioral Analysis and Anomaly Detection:** Security Information and Event Management (SIEM) systems and Intrusion Detection/Prevention Systems (IDS/IPS) should be tuned to detect anomalies. For example, a web server process suddenly spawning a **cmd.exe** or **bash** shell and performing large network transfers is a high-fidelity alert, regardless of the content's encoding.

4. Results.

When tested in a controlled lab environment simulating a basic F5 WAF deployment with common OWASP ModSecurity Core Rule Set (CRS) signatures, the multi-layer encoding technique proved highly effective. The WAF failed to block any of the exfiltration attempts.

- **Request Bypass:** Commands for directory listing, file reading, and network enumeration were successfully encoded and executed without triggering WAF blocks.
- **Response Bypass:** The exfiltration of sensitive files, including a simulated "/etc/passwd" file and a small database, was successfully hidden within the HTTP response. The WAF's data loss prevention (DLP) features, configured with static patterns, did not identify the encoded data as a threat.
- **Contrast with Plaintext:** As a control, plaintext commands and responses were immediately blocked by the WAF, confirming that the encoding was the critical bypass factor.

The results clearly demonstrate that static WAF rules are insufficient to protect against a determined attacker in the post-exploitation phase.

5. Conclusion.

The post-exploitation phase remains a critical window of opportunity for attackers and a significant challenge for defenders. As demonstrated, the static analysis performed by many traditional WAFs, including F5, can be systematically bypassed through strategic payload obfuscation. The simple yet effective technique of chaining Base64, reversal, and hex encoding provides a clear path for data exfiltration, rendering the WAF transparent.

Defense, therefore, must evolve beyond signature matching. A proactive security posture requires the implementation of dynamic analysis, strict application input validation, and robust behavioral monitoring to detect the anomalous activity that signifies a post-exploitation process, even if the communication channel itself is encrypted or encoded. The battle has shifted from merely inspecting content to understanding behavior and intent.

To further this research and explore the evolving arms race between attackers and defenders, the following avenues will be pursued:

- **Test with Custom Encryption Algorithm:** Implementing a simple, custom XOR or substitution cipher for command and data obfuscation to bypass WAFs that decode common schemes like Base64.
- **Test with Asymmetric Cryptography:** Using a public key to encrypt commands on the attacker's side, with the compromised server holding the private key for decryption. This would make the traffic impossible for the WAF to decipher without the private key.
- **Test with Remote Commanding:** Utilizing fully-featured Command and Control (C2) frameworks that natively support sophisticated traffic encoding and encryption (e.g., Meterpreter, Cobalt Strike) against updated WAFs with behavioral analytics.
- **Interactive Shell:** Developing methods to establish a fully interactive reverse shell channel where all keystrokes and responses are encrypted using the described or more advanced methods, bypassing WAFs that monitor for sustained, encrypted, or jumbled traffic patterns.

F5 WAF-ის გვერდის ავლა პოსტ-ექსპლუატაციის დროს მონაცემების ექსფილტრაციისთვის

ავთანდილ ბიჩნიგაური, ლუკა შონია, ილია შონია

საქართველოს ტექნიკური უნივერსიტეტი

bichnigauri_av@gtu.ge, shonia@gtu.ge, iliashoniafr@gmail.com

რეზიუმე

კიბერშეტევის პოსტ-ექსპლუატაციის ფაზაში კიბერდამნაშავეები აღწევენ თავიანთ საბოლოო მიზნებს, რომლებიც ხშირად ფოკუსირებულია მნიშვნელოვანი მონაცემების ექსფილტრაციაზე. მიუხედავად იმისა, რომ ვებ აპლიკაციების Firewall-ები (WAF), მაგალითად, როგორიცაა F5 Networks, კრიტიკულ თავდაცვის ფენას წარმოადგენს. მათი დამოკიდებულება სტატიკურ და ხელმოწერაზე დაფუძნებულ აღმოჩენაზე შეიძლება მნიშვნელოვანი დაუცველობის მიზეზი გახდეს. ეს სტატია იკვლევს პრაქტიკულ მეთოდოლოგიას F5 WAF დაცვის გვერდის ავლით, რათა უზრუნველყოფილი იყოს მონაცემთა ეფექტური ექსფილტრაცია საწყისი კომპრომეტირების შემდეგ. ტექნიკა მოიცავს მრავალშრიანი ენკოდირების სქემას (Base64, Reversal და Hexadecimal კოდირება), რომლებიც გამოიყენება როგორც თავდამსხმელის ბრძანებებზე HTTP მოთხოვნაში, ასევე მოპარულ მონაცემებზე HTTP პასუხში. ეს პროცესი ეფექტურად ფარავს მავნე ბრძანებებს თავდაცვის სტატიკური წესების ნაკრებებისგან. სტატიაში დეტალურადაა განხილული ეტაპობრივი მეთოდოლოგია, სტატიკური ანალიზის თანდაყოლილი შეზღუდვები WAF-ებში და წარმოდგენილია პოტენციური გადაწყვეტილებები თავდაცვისთვის, მათ შორის დინამიური ანალიზი, პარამეტრების ვალიდაცია და ქცევითი ანომალიების აღმოჩენა. დასკვნა ხაზს უსვამს ცვალებად საფრთხეთა ლანდშაფტს და გვთავაზობს სამომავლო კვლევის მიმართულებებს, როგორიცაა მორგებული დაშიფვრის და ასიმეტრიული კრიპტოგრაფიის გამოყენება მონაცემების უფრო დახვეწილი გზებით ექსფილტრაციისთვის.

საკვანძო სიტყვები: მონაცემების ექსფილტრაცია, პოსტ-ექსპლუატაცია, F5 WAF, ვებ აპლიკაციების Firewall-ის გვერდის ავლა, ენკოდირება, შიფრაცია, კიბერუსაფრთხოება, მართვა და კონტროლი, C2, სტატიკური ანალიზი, დინამიური ანალიზი, SIEM, IDS/IPS.

References:

1. A. Bichnigauri, I. Kartvelishvili, L. Shonia - “Development and implementation of a model of an effective mechanism for preventing phishing and malicious code websites in a web browser environment”, Georgian Technical University International Scientific-Practical Conference “Modern Challenges and Achievements in Information Technologies - 2023”
2. A. Bichnigauri, O. Shonia - “Means for detecting IoT devices in a local network to ensure their cybersecurity”, Scientific Works. Automated Control Systems. № 1(32), Vol.1. Tbilisi, 2021
3. A. Bichnigauri, O. Shonia, T. Kaishauri - "Detecting Suspicious Domain Names for Cyber Threat Identification Using CTL Technology", International Scientific-Practical Conference "Innovations and Modern Challenges - 2022" dedicated to the 100th anniversary of the Georgian Technical University and the 65th anniversary of the Faculty of Information Systems, Tbilisi, 2022
4. OWASP CRS: https://owasp.org/www-project-modsecurity-core-rule-set/
5. F5 Networks, "What is a Web Application Firewall (WAF)?", https://www.f5.com/glossary/web-application-firewall-waf
6. MITRE ATT&CK, "Obfuscated Files or Information - T1027", https://attack.mitre.org/techniques/T1027/
7. MITRE ATT&CK, "Exfiltration - T1020", https://attack.mitre.org/techniques/T1020/