acunetix

Affected Items Report

Acunetix Security Audit

18 February 2020

Generated by Acunetix

Scan of https://thesisbloodwebsite.000webhostapp.com/

Scan details

Scan information	
Start time	18/02/2020, 02:40:33
Start url	https://thesisbloodwebsite.000webhostapp.com/
Host	https://thesisbloodwebsite.000webhostapp.com/
Scan time	13 minutes, 11 seconds
Profile	Full Scan

Threat level

Acunetix Threat Level 3

One or more high-severity type vulnerabilities have been discovered by the scanner. A malicious user can exploit these vulnerabilities and compromise the backend database and/or deface your website.

Alerts distribution

Total alerts found	31
High	2
Medium	9
Low	8
Informational	12

Affected items

Web Server	
Alert group	**Blind SQL Injection**
Severity	High
Description	SQL injection (SQLi) refers to an injection attack wherein an attacker can execute malicious SQL statements that control a web application's database server.
Recommendations	Use parameterized queries when dealing with SQL queries that contains user input. Parameterized queries allows the database to understand which parts of the SQL query should be considered as user input, therefore solving SQL injection.
Alert variants	
Details	Not available in the free trial
`Not available in the free trial`	

Web Server	
Alert group	**Cross site scripting**
Severity	High
Description	Cross-site Scripting (XSS) refers to client-side code injection attack wherein an attacker can execute malicious scripts into a legitimate website or web application. XSS occurs when a web application makes use of unvalidated or unencoded user input within the output it generates.
Recommendations	Apply context-dependent encoding and/or validation to user input rendered on a page
Alert variants	
Details	Not available in the free trial
`Not available in the free trial`	

Web Server	
Alert group	**Application error message**
Severity	Medium
Description	This alert requires manual confirmation Application error or warning messages may expose sensitive information about an application's internal workings to an attacker. Acunetix found an error or warning message that may disclose sensitive information. The message may also contain the location of the file that produced an unhandled exception. Consult the 'Attack details' section for more information about the affected page.
Recommendations	Verify that this page is disclosing error or warning messages and properly configure the application to log errors to a file instead of displaying the error to the user.
Alert variants	
Details	Not available in the free trial
`Not available in the free trial`	

Web Server	
Alert group	**Application error message**
Severity	Medium
Description	This alert requires manual confirmation Application error or warning messages may expose sensitive information about an application's internal workings to an attacker. Acunetix found an error or warning message that may disclose sensitive information. The message may also contain the location of the file that produced an unhandled exception. Consult the 'Attack details' section for more information about the affected page.
Recommendations	Verify that this page is disclosing error or warning messages and properly configure the application to log errors to a file instead of displaying the error to the user.

Alert variants	
Details	Not available in the free trial
Not available in the free trial	

Web Server	
Alert group	**Directory listing**
Severity	Medium
Description	The web server is configured to display the list of files contained in this directory. This is not recommended because the directory may contain files that are not normally exposed through links on the web site.
Recommendations	You should make sure the directory does not contain sensitive information or you may want to restrict directory listings from the web server configuration.
Alert variants	
Details	Not available in the free trial
Not available in the free trial	

Web Server	
Alert group	**Error message on page**
Severity	Medium
Description	This alert requires manual confirmation Application error or warning messages may expose sensitive information about an application's internal workings to an attacker. Acunetix found an error or warning message that may disclose sensitive information. The message may also contain the location of the file that produced an unhandled exception. Consult the 'Attack details' section for more information about the affected page.
Recommendations	Verify that this page is disclosing error or warning messages and properly configure the application to log errors to a file instead of displaying the error to the user.
Alert variants	
Details	Not available in the free trial
Not available in the free trial	

Web Server	
Alert group	**HTML form without CSRF protection**
Severity	Medium
Description	This alert requires manual confirmation Cross-Site Request Forgery (CSRF, or XSRF) is a vulnerability wherein an attacker tricks a victim into making a request the victim did not intend to make. Therefore, with CSRF, an attacker abuses the trust a web application has with a victim's browser. Acunetix found an HTML form with no apparent anti-CSRF protection implemented. Consult the 'Attack details' section for more information about the affected HTML form.
Recommendations	Verify if this form requires anti-CSRF protection and implement CSRF countermeasures if necessary. The recommended and the most widely used technique for preventing CSRF attacks is know as an anti-CSRF token, also sometimes referred to as a synchronizer token. The characteristics of a well designed anti-CSRF system involve the following attributes. • The anti-CSRF token should be unique for each user session • The session should automatically expire after a suitable amount of time • The anti-CSRF token should be a cryptographically random value of significant length • The anti-CSRF token should be cryptographically secure, that is, generated by a strong Pseudo-Random Number Generator (PRNG) algorithm • The anti-CSRF token is added as a hidden field for forms, or within URLs (only necessary if

	GET requests cause state changes, that is, GET requests are not idempotent) • The server should reject the requested action if the anti-CSRF token fails validation When a user submits a form or makes some other authenticated request that requires a Cookie, the anti-CSRF token should be included in the request. Then, the web application will then verify the existence and correctness of this token before processing the request. If the token is missing or incorrect, the request can be rejected.
Alert variants	
Details	Not available in the free trial
`Not available in the free trial`	

Web Server	
Alert group	**Source code disclosure**
Severity	Medium
Description	Looks like the source code for this script is available. This check is using pattern matching to determine if server side tags are found in the file. In some cases this alert may generate false positives.
Recommendations	Remove this file from your website or change its permissions to remove access.
Alert variants	
Details	Not available in the free trial
`Not available in the free trial`	

Web Server	
Alert group	**Source code disclosure**
Severity	Medium
Description	Looks like the source code for this script is available. This check is using pattern matching to determine if server side tags are found in the file. In some cases this alert may generate false positives.
Recommendations	Remove this file from your website or change its permissions to remove access.
Alert variants	
Details	Not available in the free trial
`Not available in the free trial`	

Web Server	
Alert group	**Source code disclosure**
Severity	Medium
Description	Looks like the source code for this script is available. This check is using pattern matching to determine if server side tags are found in the file. In some cases this alert may generate false positives.
Recommendations	Remove this file from your website or change its permissions to remove access.
Alert variants	
Details	Not available in the free trial
`Not available in the free trial`	

Web Server	
Alert group	**Source code disclosure**
Severity	Medium
Description	Looks like the source code for this script is available. This check is using pattern matching to determine if server side tags are found in the file. In some cases this alert may generate false positives.
Recommendations	Remove this file from your website or change its permissions to remove access.
Alert variants	

Details	Not available in the free trial
Not available in the free trial	

Web Server	
Alert group	**Clickjacking: X-Frame-Options header missing**
Severity	Low
Description	Clickjacking (User Interface redress attack, UI redress attack, UI redressing) is a malicious technique of tricking a Web user into clicking on something different from what the user perceives they are clicking on, thus potentially revealing confidential information or taking control of their computer while clicking on seemingly innocuous web pages. The server didn't return an **X-Frame-Options** header which means that this website could be at risk of a clickjacking attack. The X-Frame-Options HTTP response header can be used to indicate whether or not a browser should be allowed to render a page inside a frame or iframe. Sites can use this to avoid clickjacking attacks, by ensuring that their content is not embedded into other sites.
Recommendations	Configure your web server to include an X-Frame-Options header. Consult Web references for more information about the possible values for this header.
Alert variants	
Details	Not available in the free trial
Not available in the free trial	

Web Server	
Alert group	**Cookie(s) without HttpOnly flag set**
Severity	Low
Description	This cookie does not have the HTTPOnly flag set. When a cookie is set with the HTTPOnly flag, it instructs the browser that the cookie can only be accessed by the server and not by client-side scripts. This is an important security protection for session cookies.
Recommendations	If possible, you should set the HTTPOnly flag for this cookie.
Alert variants	
Details	Not available in the free trial
Not available in the free trial	

Web Server	
Alert group	**Cookie(s) without Secure flag set**
Severity	Low
Description	This cookie does not have the Secure flag set. When a cookie is set with the Secure flag, it instructs the browser that the cookie can only be accessed over secure SSL channels. This is an important security protection for session cookies.
Recommendations	If possible, you should set the Secure flag for this cookie.
Alert variants	
Details	Not available in the free trial
Not available in the free trial	

Web Server	
Alert group	**HTML Form found in redirect page**
Severity	Low
	Manual confirmation is required for this alert. An HTML form was found in the response body of this page. However, the current page redirects the visitor to another page by returning an HTTP status code of 301/302. Therefore, all browser users will not see the contents of this page and will not be able to interact with the HTML form. Sometimes programmers don't properly terminate the script after redirecting the user to another page. For example:

```
<?php

    if (!isset($_SESSION["authenticated"])) {

        header("Location: auth.php");

    }

?>

<title>Administration page</title>

<form action="/admin/action" method="post">

    <!-- ...  form inputs ...  -->

</form>



<!-- ...  the rest of the administration page ...  -->
```

Description

This script is incorrect because the script is not terminated after the "header("Location: auth.php");" line. An attacker can access the content the administration page by using an HTTP client that doesn't follow redirection (like HTTP Editor). This creates an authentication bypass vulnerability.
The correct code would be

```
<?php

    if (!isset($_SESSION[auth])) {

        header("Location: auth.php");

        exit();

    }

?>

<title>Administration page</title>

<form action="/admin/action" method="post">

    <!-- ...  form inputs ...  -->

</form>



<!-- ...  the rest of the administration page ...  -->
```

Recommendations	Make sure the script is terminated after redirecting the user to another page.
Alert variants	
Details	Not available in the free trial
Not available in the free trial	

Web Server	

Alert group	**HTML Form found in redirect page**
Severity	Low

Description

Manual confirmation is required for this alert.

An HTML form was found in the response body of this page. However, the current page redirects the visitor to another page by returning an HTTP status code of 301/302. Therefore, all browser users will not see the contents of this page and will not be able to interact with the HTML form.

Sometimes programmers don't properly terminate the script after redirecting the user to another page. For example:

```
<?php

    if (!isset($_SESSION["authenticated"])) {

        header("Location: auth.php");

    }

?>

<title>Administration page</title>

<form action="/admin/action" method="post">

    <!-- ...  form inputs ...  -->

</form>



<!-- ...  the rest of the administration page ...  -->
```

This script is incorrect because the script is not terminated after the "header("Location: auth.php");" line. An attacker can access the content the administration page by using an HTTP client that doesn't follow redirection (like HTTP Editor). This creates an authentication bypass vulnerability.
The correct code would be

```
<?php
    if (!isset($_SESSION[auth])) {
        header("Location: auth.php");
        exit();
    }
?>
<title>Administration page</title>
<form action="/admin/action" method="post">
    <!-- ...  form inputs ...  -->
</form>


<!-- ...  the rest of the administration page ...  -->
```

Recommendations	Make sure the script is terminated after redirecting the user to another page.
Alert variants	
Details	Not available in the free trial

```
Not available in the free trial
```

Web Server	
Alert group	**HTML Form found in redirect page**
Severity	Low

Manual confirmation is required for this alert.

An HTML form was found in the response body of this page. However, the current page redirects the visitor to another page by returning an HTTP status code of 301/302. Therefore, all browser users will not see the contents of this page and will not be able to interact with the HTML form.

Sometimes programmers don't properly terminate the script after redirecting the user to another page. For example:

Description

```
<?php
    if (!isset($_SESSION["authenticated"])) {
        header("Location: auth.php");
    }
?>
<title>Administration page</title>
<form action="/admin/action" method="post">
    <!-- ...  form inputs ...  -->
</form>


<!-- ...  the rest of the administration page ...  -->
```

This script is incorrect because the script is not terminated after the "header("Location: auth.php");" line. An attacker can access the content the administration page by using an HTTP client that doesn't follow redirection (like HTTP Editor). This creates an authentication bypass vulnerability.
The correct code would be

```
<?php
    if (!isset($_SESSION[auth])) {
        header("Location: auth.php");
        exit();
    }
?>
<title>Administration page</title>
<form action="/admin/action" method="post">
    <!-- ...  form inputs ...  -->
</form>


<!-- ...  the rest of the administration page ...  -->
```

Recommendations	Make sure the script is terminated after redirecting the user to another page.
Alert variants	
Details	Not available in the free trial

```
Not available in the free trial
```

Web Server	

<table>
<tr><th>Alert group</th><th>HTML Form found in redirect page</th></tr>
<tr><td>Severity</td><td>Low</td></tr>
<tr><td>Description</td><td>

Manual confirmation is required for this alert.

An HTML form was found in the response body of this page. However, the current page redirects the visitor to another page by returning an HTTP status code of 301/302. Therefore, all browser users will not see the contents of this page and will not be able to interact with the HTML form.

Sometimes programmers don't properly terminate the script after redirecting the user to another page. For example:

```
<?php

    if (!isset($_SESSION["authenticated"])) {

        header("Location: auth.php");

    }

?>

<title>Administration page</title>

<form action="/admin/action" method="post">

    <!-- ...  form inputs ...  -->

</form>



<!-- ...  the rest of the administration page ...  -->
```

This script is incorrect because the script is not terminated after the "header("Location: auth.php");" line. An attacker can access the content the administration page by using an HTTP client that doesn't follow redirection (like HTTP Editor). This creates an authentication bypass vulnerability.
The correct code would be

</td></tr>
</table>

```
<?php
    if (!isset($_SESSION[auth])) {
        header("Location: auth.php");
        exit();
    }
?>
<title>Administration page</title>
<form action="/admin/action" method="post">
    <!-- ...  form inputs ...  -->
</form>


<!-- ...  the rest of the administration page ...  -->
```

Recommendations	Make sure the script is terminated after redirecting the user to another page.
Alert variants	
Details	Not available in the free trial

```
Not available in the free trial
```

Web Server	
Alert group	**HTML Form found in redirect page**
Severity	Low

Manual confirmation is required for this alert.

An HTML form was found in the response body of this page. However, the current page redirects the visitor to another page by returning an HTTP status code of 301/302. Therefore, all browser users will not see the contents of this page and will not be able to interact with the HTML form.

Sometimes programmers don't properly terminate the script after redirecting the user to another page. For example:

Description

```
<?php
    if (!isset($_SESSION["authenticated"])) {
        header("Location: auth.php");
    }
?>
<title>Administration page</title>
<form action="/admin/action" method="post">
    <!-- ...  form inputs ...  -->
</form>


<!-- ...  the rest of the administration page ...  -->
```

This script is incorrect because the script is not terminated after the "header("Location: auth.php");" line. An attacker can access the content the administration page by using an HTTP client that doesn't follow redirection (like HTTP Editor). This creates an authentication bypass vulnerability.
The correct code would be

```
<?php
    if (!isset($_SESSION[auth])) {
        header("Location: auth.php");
        exit();
    }
?>
<title>Administration page</title>
<form action="/admin/action" method="post">
    <!-- ...  form inputs ...  -->
</form>


<!-- ...  the rest of the administration page ...  -->
```

Recommendations	Make sure the script is terminated after redirecting the user to another page.
Alert variants	
Details	Not available in the free trial

```
Not available in the free trial
```

Web Server	

Alert group	**Email address found**
Severity	Informational
Description	One or more email addresses have been found on this page. The majority of spam comes from email addresses harvested off the internet. The spam-bots (also known as email harvesters and email extractors) are programs that scour the internet looking for email addresses on any website they come across. Spambot programs look for strings like myname@mydomain.com and then record any addresses found.
Recommendations	Check references for details on how to solve this problem.
Alert variants	
Details	Not available in the free trial
Not available in the free trial	

Web Server	
Alert group	**Email address found**
Severity	Informational
Description	One or more email addresses have been found on this page. The majority of spam comes from email addresses harvested off the internet. The spam-bots (also known as email harvesters and email extractors) are programs that scour the internet looking for email addresses on any website they come across. Spambot programs look for strings like myname@mydomain.com and then record any addresses found.
Recommendations	Check references for details on how to solve this problem.
Alert variants	
Details	Not available in the free trial
Not available in the free trial	

Web Server	
Alert group	**Email address found**
Severity	Informational
Description	One or more email addresses have been found on this page. The majority of spam comes from email addresses harvested off the internet. The spam-bots (also known as email harvesters and email extractors) are programs that scour the internet looking for email addresses on any website they come across. Spambot programs look for strings like myname@mydomain.com and then record any addresses found.
Recommendations	Check references for details on how to solve this problem.
Alert variants	
Details	Not available in the free trial
Not available in the free trial	

Web Server	
Alert group	**Email address found**
Severity	Informational
Description	One or more email addresses have been found on this page. The majority of spam comes from email addresses harvested off the internet. The spam-bots (also known as email harvesters and email extractors) are programs that scour the internet looking for email addresses on any website they come across. Spambot programs look for strings like myname@mydomain.com and then record any addresses found.
Recommendations	Check references for details on how to solve this problem.
Alert variants	
Details	Not available in the free trial
Not available in the free trial	

Web Server	
Alert group	**Email address found**

Severity	Informational
Description	One or more email addresses have been found on this page. The majority of spam comes from email addresses harvested off the internet. The spam-bots (also known as email harvesters and email extractors) are programs that scour the internet looking for email addresses on any website they come across. Spambot programs look for strings like myname@mydomain.com and then record any addresses found.
Recommendations	Check references for details on how to solve this problem.
Alert variants	
Details	Not available in the free trial
`Not available in the free trial`	

Web Server	
Alert group	**Email address found**
Severity	Informational
Description	One or more email addresses have been found on this page. The majority of spam comes from email addresses harvested off the internet. The spam-bots (also known as email harvesters and email extractors) are programs that scour the internet looking for email addresses on any website they come across. Spambot programs look for strings like myname@mydomain.com and then record any addresses found.
Recommendations	Check references for details on how to solve this problem.
Alert variants	
Details	Not available in the free trial
`Not available in the free trial`	

Web Server	
Alert group	**Email address found**
Severity	Informational
Description	One or more email addresses have been found on this page. The majority of spam comes from email addresses harvested off the internet. The spam-bots (also known as email harvesters and email extractors) are programs that scour the internet looking for email addresses on any website they come across. Spambot programs look for strings like myname@mydomain.com and then record any addresses found.
Recommendations	Check references for details on how to solve this problem.
Alert variants	
Details	Not available in the free trial
`Not available in the free trial`	

Web Server	
Alert group	**Email address found**
Severity	Informational
Description	One or more email addresses have been found on this page. The majority of spam comes from email addresses harvested off the internet. The spam-bots (also known as email harvesters and email extractors) are programs that scour the internet looking for email addresses on any website they come across. Spambot programs look for strings like myname@mydomain.com and then record any addresses found.
Recommendations	Check references for details on how to solve this problem.
Alert variants	
Details	Not available in the free trial
`Not available in the free trial`	

Web Server	
Alert group	**Email address found**
Severity	Informational

Description	One or more email addresses have been found on this page. The majority of spam comes from email addresses harvested off the internet. The spam-bots (also known as email harvesters and email extractors) are programs that scour the internet looking for email addresses on any website they come across. Spambot programs look for strings like myname@mydomain.com and then record any addresses found.
Recommendations	Check references for details on how to solve this problem.
Alert variants	
Details	Not available in the free trial
`Not available in the free trial`	

Web Server	
Alert group	**Email address found**
Severity	Informational
Description	One or more email addresses have been found on this page. The majority of spam comes from email addresses harvested off the internet. The spam-bots (also known as email harvesters and email extractors) are programs that scour the internet looking for email addresses on any website they come across. Spambot programs look for strings like myname@mydomain.com and then record any addresses found.
Recommendations	Check references for details on how to solve this problem.
Alert variants	
Details	Not available in the free trial
`Not available in the free trial`	

Web Server	
Alert group	**Email address found**
Severity	Informational
Description	One or more email addresses have been found on this page. The majority of spam comes from email addresses harvested off the internet. The spam-bots (also known as email harvesters and email extractors) are programs that scour the internet looking for email addresses on any website they come across. Spambot programs look for strings like myname@mydomain.com and then record any addresses found.
Recommendations	Check references for details on how to solve this problem.
Alert variants	
Details	Not available in the free trial
`Not available in the free trial`	

Web Server	
Alert group	**Password type input with auto-complete enabled**
Severity	Informational
Description	When a new name and password is entered in a form and the form is submitted, the browser asks if the password should be saved.Thereafter when the form is displayed, the name and password are filled in automatically or are completed as the name is entered. An attacker with local access could obtain the cleartext password from the browser cache.
Recommendations	The password auto-complete should be disabled in sensitive applications. To disable auto-complete, you may use a code similar to: `<INPUT TYPE="password" AUTOCOMPLETE="off">`
Alert variants	
Details	Not available in the free trial
`Not available in the free trial`	

Scanned items (coverage report)

https://thesisbloodwebsite.000webhostapp.com/
https://thesisbloodwebsite.000webhostapp.com/admin.php
https://thesisbloodwebsite.000webhostapp.com/bloodlist
https://thesisbloodwebsite.000webhostapp.com/bloodlist/aheader.php
https://thesisbloodwebsite.000webhostapp.com/bloodlist/Apositive.php
https://thesisbloodwebsite.000webhostapp.com/bloodlist/logout.php
https://thesisbloodwebsite.000webhostapp.com/donor-list.php
https://thesisbloodwebsite.000webhostapp.com/donor-reg.php
https://thesisbloodwebsite.000webhostapp.com/entry.php
https://thesisbloodwebsite.000webhostapp.com/icons
https://thesisbloodwebsite.000webhostapp.com/index.php
https://thesisbloodwebsite.000webhostapp.com/logout.php
https://thesisbloodwebsite.000webhostapp.com/ngo-list.php
https://thesisbloodwebsite.000webhostapp.com/out-stock-blood-list.php
https://thesisbloodwebsite.000webhostapp.com/stock-blood-list.php