Web Vulnerability Scanners: A Case Study

Angel Rajan, Emre Erturk

Eastern Institute of Technology, Hawke's Bay

Abstract

Cloud security is one of the biggest concerns for many companies. The growth in the number and size of websites increases the need for better securing those websites. Manual testing and detection of web vulnerabilities can be very time consuming. Automated Web Vulnerability Scanners (WVS) help with the detection of vulnerabilities in web applications. Acunetix is one of the widely used vulnerability scanners. Acunetix is also easy to implement and to use. The scan results not only provide the details of the vulnerabilities, but also give information about fixing the vulnerabilities. AcuSensor and AcuMonitor (technologies used by Acunetix) help generate more accurate potential vulnerability results. One of the purposes of this paper is to orient current students of computer security with using vulnerability scanners. Secondly, this paper provides a literature review related to the topic of security vulnerability scanners. Finally, web vulnerabilities are addressed from the mobile device and browser perspectives.

Introduction

With the advancements in cloud computing, web services, and browser based applications, most business rely on conducting their business communications and transactions online. However, these websites and web applications are not completely secure. Around 30,000 websites are being attacked every day (Lyne, 2013), and one out of every three websites is vulnerable to hacking (Schupak, 2015). Moreover, ninety percent of passwords are vulnerable to being stolen (Warman, 2013). The increasing number of websites and online applications increases the urgency of securing these websites.

Web security scanners are automated tools that check out websites or web applications for security vulnerabilities, without accessing the application's source code (Saeed, 2014). Web vulnerability scanners help to find vulnerabilities of web applications and websites. A security vulnerability is a weakness that may be exploited to cause damage, but its presence does not cause harm by itself (Jeeva, Raveena, Sangeetha, & Vinothini, 2016). Grabber, Vega, Acunetix, Wapiti (InfoSec Institute, 2014) are few examples of web vulnerability scanners.

The Cloud Security Alliance (2016) has recently identified twelve major types of security concerns and threats. Many of these are relevant to areas where web vulnerability scanners may be helpful in reducing risks. For example, two of these concerns, insecure APIs and insufficient due diligence, may be overlooked by web developers and web masters.

This report focuses on Acunetix, one of the widely used web vulnerability scanner. It begins with a description of Acunetix with the details of Acunetix used for this case study. Then the report tries to explain the working of Acunetix with proper screenshots. In the next section, few high priority vulnerabilities identified by Acunetix are described. AcuSensor and AcuMonitor technologies implemented by Acunetix is also discussed. Advantages and disadvantages of Acunetix is discussed in the next section and finally this report finishes with some recommendations and conclusion.

Case Study: Acunetix

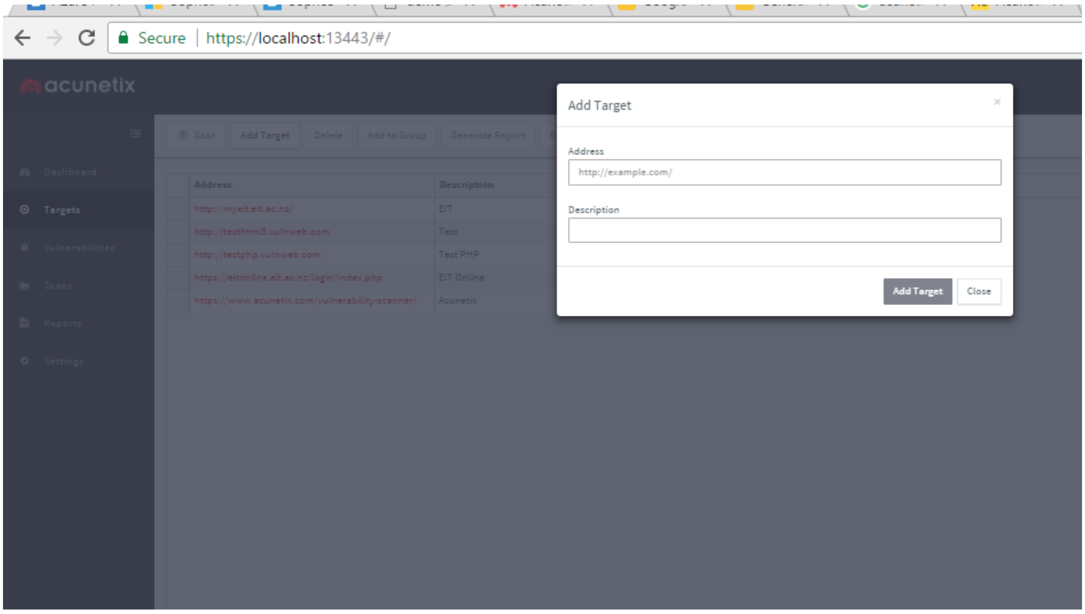
Acunetix is an automated web vulnerability scanner which scans any web application or websites that use HTTP or HTTPS protocols and are accessible through a web browser. It audits the websites by identifying vulnerabilities, such as SQL injection, cross site scripting, and others.

The following table (Table 1) gives the description of the version, cost and other details of Acunetix used for this paper. Websites mainly used in the walkthrough are test websites as opposed to organisational websites. Acunetix is available in four versions, online, standard, pro and enterprise.

Table 1. *Version and platform of Acunetix used*

Version	Acunetix 11 Trial
Platform	Windows 8
Cost	Free
Trial period	14 days trial
Test websites	http://testasp.vulnweb.com/, http://testhtml5.vulnweb.com, http://testphp.vulnweb.com

The address of a website or web application that needs to be scanned should be added to the target. A description of the website can also be given while adding the target. Figure 1 below, is the screenshot of adding a target with description.



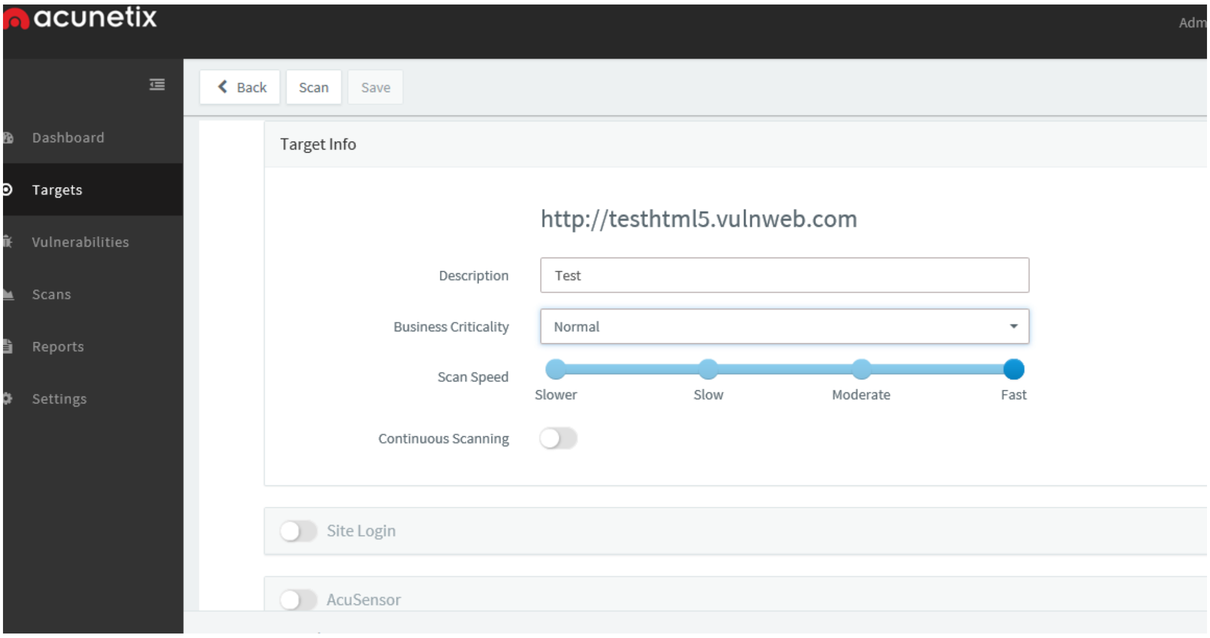
Address	Description
http://myeit.eit.ac.nz/	EIT
http://testhtml5.vulnweb.com	Test
http://testphp.vulnweb.com	Test PHP
https://eitonline.eit.ac.nz/login/index.php	EIT Online
https://www.acunetix.com/vulnerability-scanner/	Acunetix

Address: http://example.com/

Description:

Figure 1. Screenshot of adding target.

After adding the target, website is ready for scanning. As seen in Figure 2, options for setting the business criticality, scan speed and others are available. Websites and web applications can also be kept for continuous scanning.



http://testhtml5.vulnweb.com

Description: Test

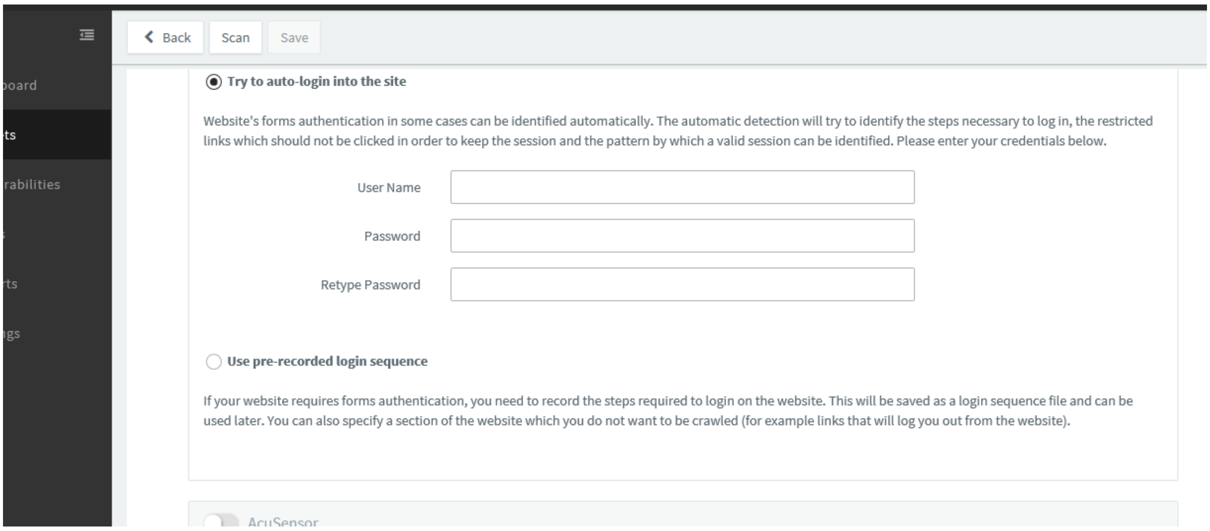
Business Criticality: Normal

Scan Speed: Fast

Continuous Scanning: ○

Figure 2. Screenshot of setting target information.

As shown in Figure 3, Acunetix offers a feature which tries to auto login to the targeted website. To accommodate this feature, two options are available. The tester can either enter the username and password manually, or a use a pre-recorded login sequence for auto login.



● Try to auto-login into the site

Website's forms authentication in some cases can be identified automatically. The automatic detection will try to identify the steps necessary to log in, the restricted links which should not be clicked in order to keep the session and the pattern by which a valid session can be identified. Please enter your credentials below.

User Name

Password

Retype Password

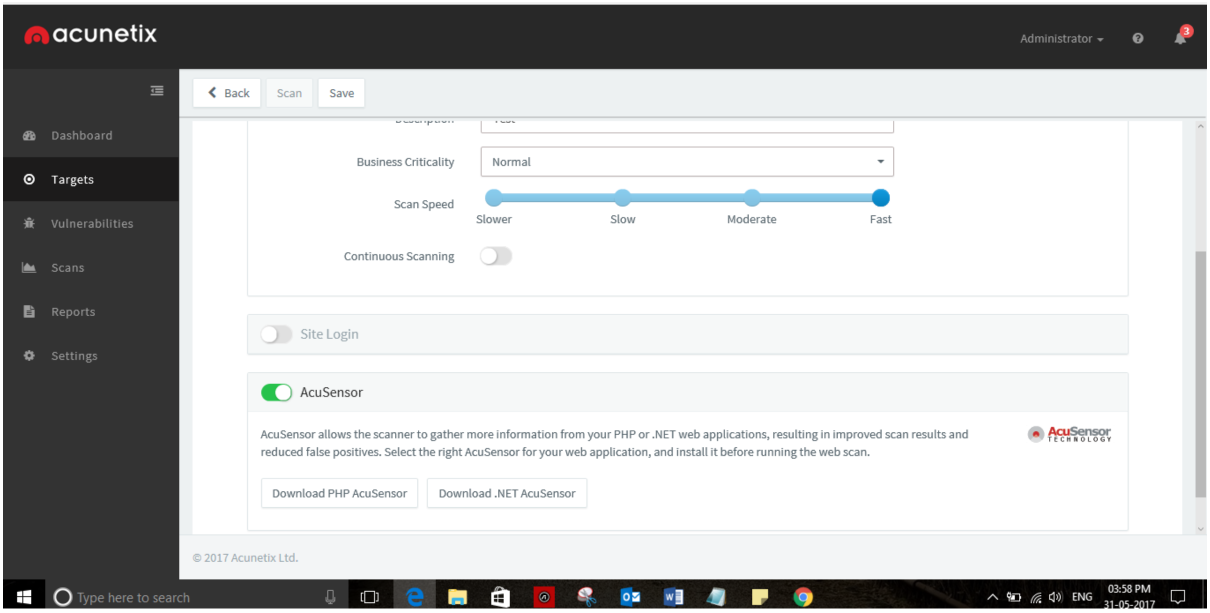
○ Use pre-recorded login sequence

If your website requires forms authentication, you need to record the steps required to login on the website. This will be saved as a login sequence file and can be used later. You can also specify a section of the website which you do not want to be crawled (for example links that will log you out from the website).

AcuSensor

Figure 3. Screenshot of setting auto login option.

If the targeted web application is using PHP or .NET, the scan results can be improved by downloading and installing the proper AcuSensor. Figure 4 below shows enabling of AcuSensor.



Business Criticality: Normal

Scan Speed: Slower — Slow — Moderate — Fast ●

Continuous Scanning: off

Site Login: off

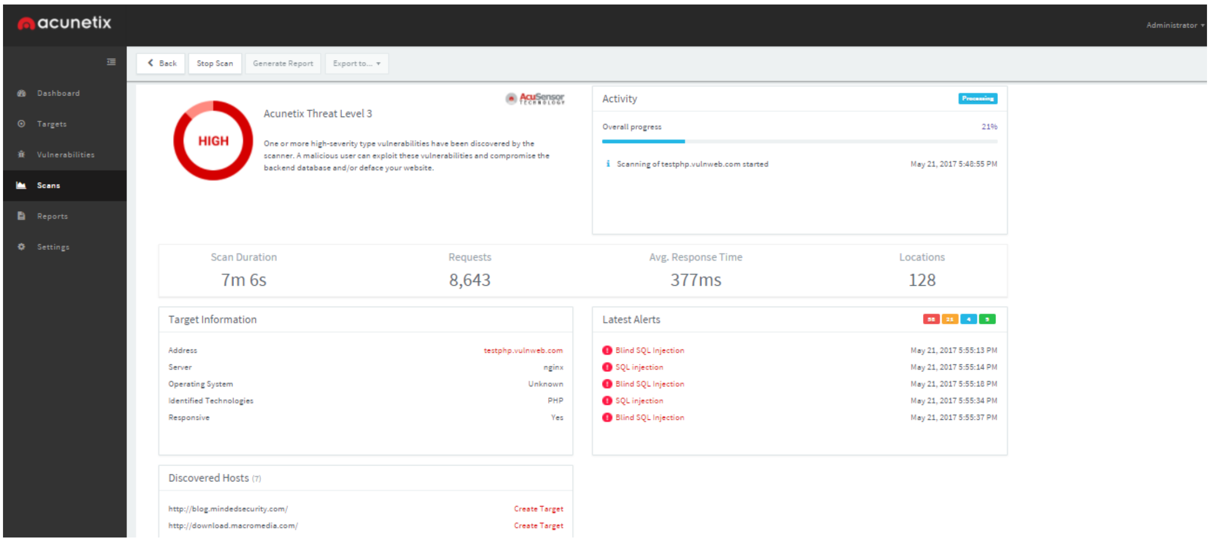
AcuSensor: on

AcuSensor allows the scanner to gather more information from your PHP or .NET web applications, resulting in improved scan results and reduced false positives. Select the right AcuSensor for your web application, and install it before running the web scan.

© 2017 Acunetix Ltd.

Figure 4. Screenshot of enabling AcuSensor.

During scanning, Acunetix provides details such as scan progress, scan duration, number of requests sent, average response time, and information about the target. It also provides the latest detected vulnerabilities and their priorities. Based on the detected vulnerabilities, Acunetix gives the overall threat level of website or web application. Figure 5 shows a screenshot of Acunetix during a scan.



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.

Activity

Overall progress 21%

Scanning of testphp.vulnweb.com started May 21, 2017 5:48:55 PM

Scan Duration	Requests	Avg. Response Time	Locations
7m 6s	8,643	377ms	128

Target Information

Address	testphp.vulnweb.com
Server	nginx
Operating System	Unknown
Identified Technologies	PHP
Responsive	Yes

Latest Alerts

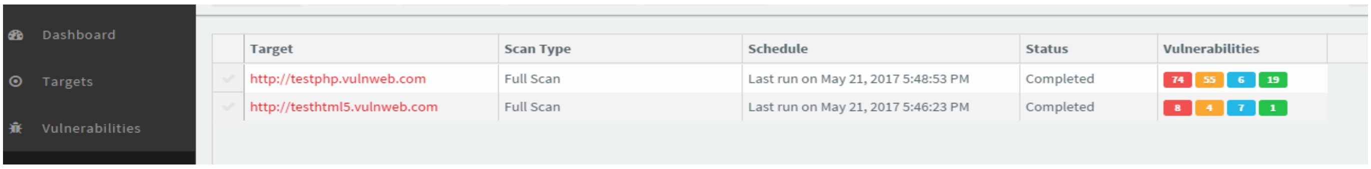
Blind SQL Injection	May 21, 2017 5:55:13 PM
SQL injection	May 21, 2017 5:55:14 PM
Blind SQL Injection	May 21, 2017 5:55:18 PM
SQL injection	May 21, 2017 5:55:34 PM
Blind SQL Injection	May 21, 2017 5:55:37 PM

Discovered Hosts (7)

http://blog.mindedsecurity.com/	Create Target
http://download.macromedia.com/	Create Target

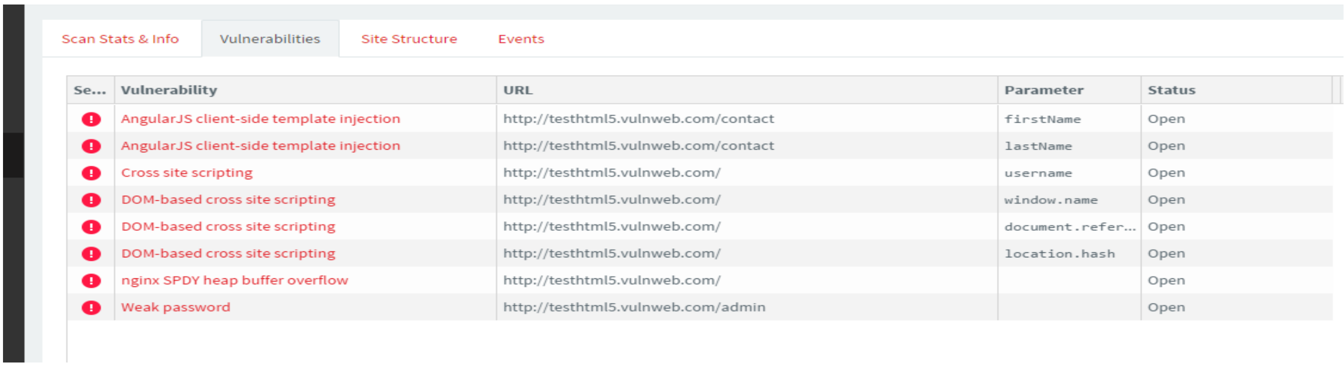
Figure 5. Screenshot of scanning website.

Once scanning is finished, the vulnerabilities detected by Acunetix are listed, based on the priority. It gives the name, URL, parameter, and status of the threat detected. Figure 6 shows the screenshot of scan result and Figure 7 shows the vulnerabilities identified during the scan.



Target	Scan Type	Schedule	Status	Vulnerabilities
http://testphp.vulnweb.com	Full Scan	Last run on May 21, 2017 5:48:53 PM	Completed	74 55 6 19
http://testhtml5.vulnweb.com	Full Scan	Last run on May 21, 2017 5:46:23 PM	Completed	8 4 7 1

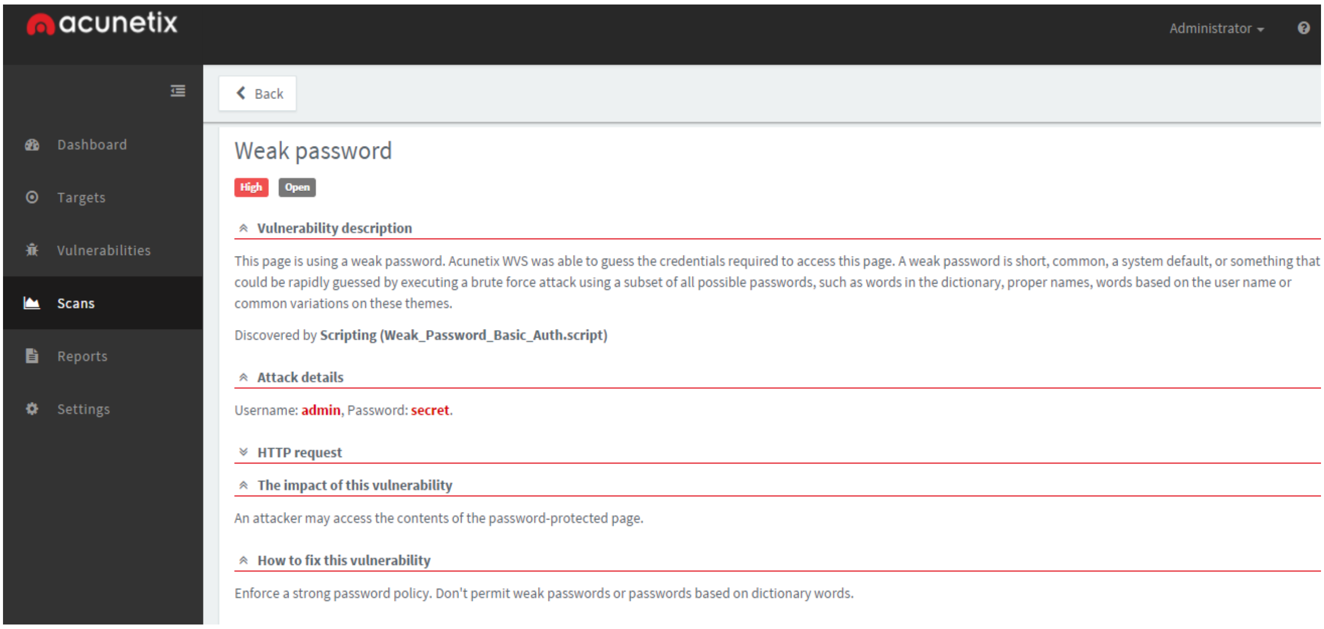
Figure 6. Screenshot of scan result.



Se...	Vulnerability	URL	Parameter	Status
!	AngularJS client-side template injection	http://testhtml5.vulnweb.com/contact	firstName	Open
!	AngularJS client-side template injection	http://testhtml5.vulnweb.com/contact	lastName	Open
!	Cross site scripting	http://testhtml5.vulnweb.com/	username	Open
!	DOM-based cross site scripting	http://testhtml5.vulnweb.com/	window.name	Open
!	DOM-based cross site scripting	http://testhtml5.vulnweb.com/	document.refer...	Open
!	DOM-based cross site scripting	http://testhtml5.vulnweb.com/	location.hash	Open
!	nginx SPDY heap buffer overflow	http://testhtml5.vulnweb.com/		Open
!	Weak password	http://testhtml5.vulnweb.com/admin		Open

Figure 7. Screenshot of vulnerabilities.

Each vulnerability detected provides additional description, impact and useful tips to fix the vulnerability. It also provides the HTTP request sent, which can help in fixing and testing the particular vulnerability. Figure 8 below shows the description and impact of vulnerability "weak password".



Weak password

High Open

Vulnerability description

This page is using a weak password. Acunetix WVS was able to guess the credentials required to access this page. A weak password is short, common, a system default, or something that could be rapidly guessed by executing a brute force attack using a subset of all possible passwords, such as words in the dictionary, proper names, words based on the user name or common variations on these themes.

Discovered by **Scripting (Weak_Password_Basic_Auth.script)**

Attack details

Username: **admin**, Password: **secret**.

HTTP request

The impact of this vulnerability

An attacker may access the contents of the password-protected page.

How to fix this vulnerability

Enforce a strong password policy. Don't permit weak passwords or passwords based on dictionary words.

Figure 8. Screenshot of vulnerability description.

Scan reports can be generated based on the scan results. Reports are available in different templates, such as affected items, developer, executive summary, and also several compliance reports can be generated. All the reports generated will be available under the reports tab, and can be downloaded later. Figure 9 shows the options for generating reports.

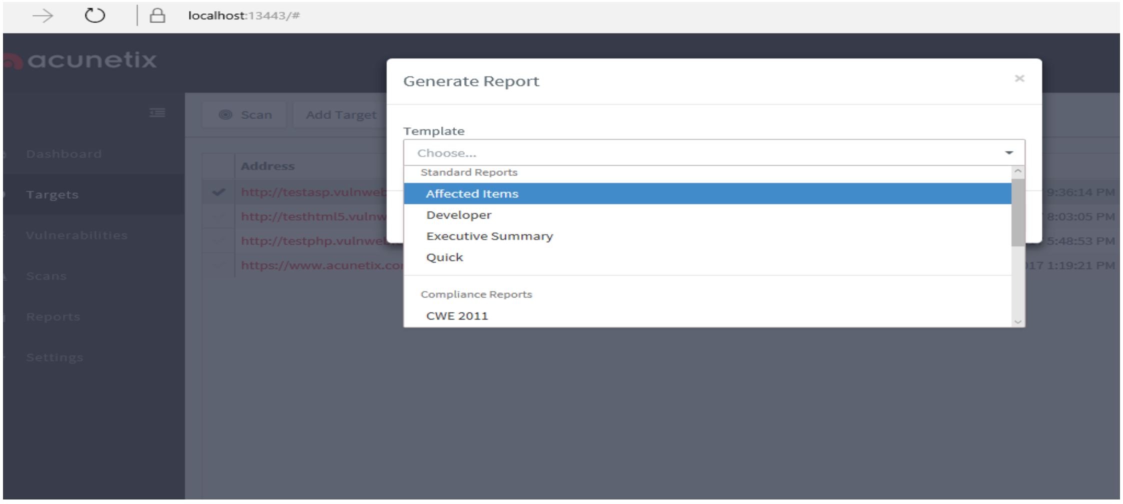
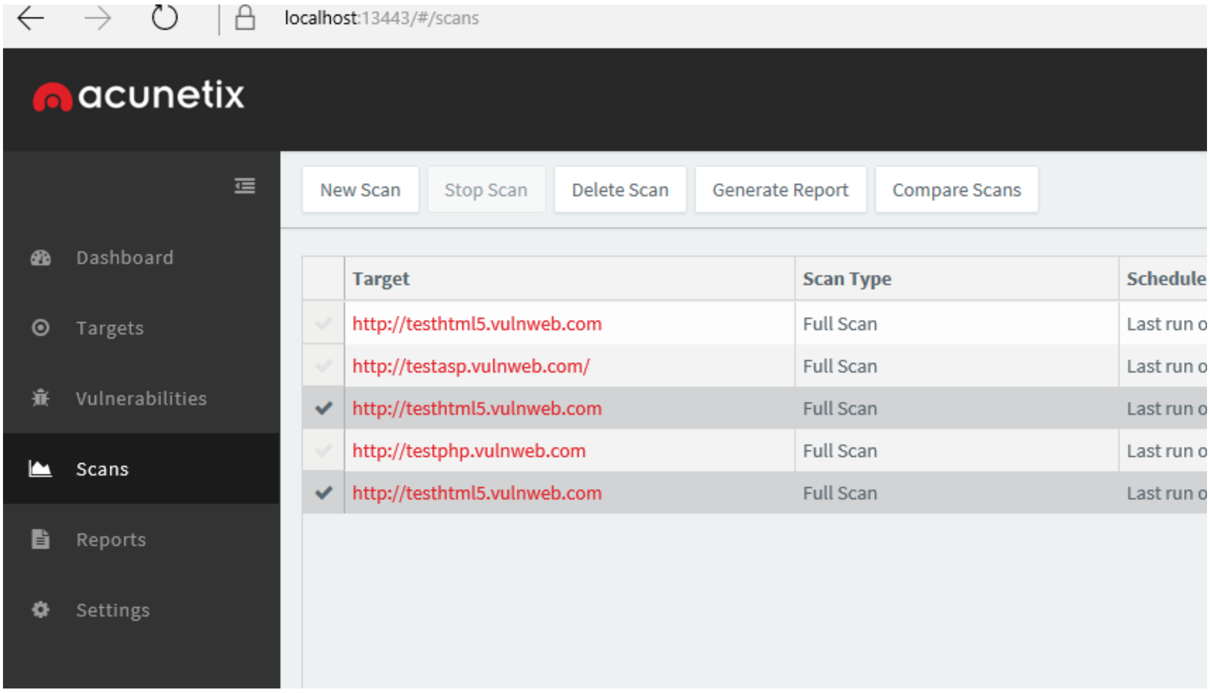


Figure 9. Screenshot of selecting report template.

Two scans for a single target can be compared and a comparison report is generated by Acunetix. This can be helpful to identify whether the fixes for the threats are working correctly and if those fixes do not induce more vulnerabilities. Figure 10 below shows selection of two scans on same target and "compare scans" button gets enabled.



	Target	Scan Type	Schedule
✓	http://testhtml5.vulnweb.com	Full Scan	Last run o
✓	http://testasp.vulnweb.com/	Full Scan	Last run o
✓	http://testhtml5.vulnweb.com	Full Scan	Last run o
✓	http://testphp.vulnweb.com	Full Scan	Last run o
✓	http://testhtml5.vulnweb.com	Full Scan	Last run o

Figure 10. Screenshot of comparing scan results.

Vulnerabilities

Few high priority vulnerabilities detected by Acunetix include Cross Site Scripting (XSS), SQL injection, Blind SQL injection, and directory traversal.

XSS is inserting malicious code into a victim's web application so that, when a victim browses the web application, the malicious script code is executed (Gupta & Gupta, 2015). A hacker injects malicious codes in the dynamic websites and when the code is executed in the web browser, it changes the web pages (Jasmine, Devi, & George, 2017). The goal of an XSS attack is to get access to the client cookies or any other sensitive information, which is used to authenticate the client to the website (Jasmine, Devi, & George, 2017).

When users visit a website, their browsers send HTTP requests, in which the headers include information about their browsers and operating systems. Based on this information, the users may be directed to the mobile version of website that, along with different content, may have different vulnerabilities. This has significant implications while trying to identify XSS issues.

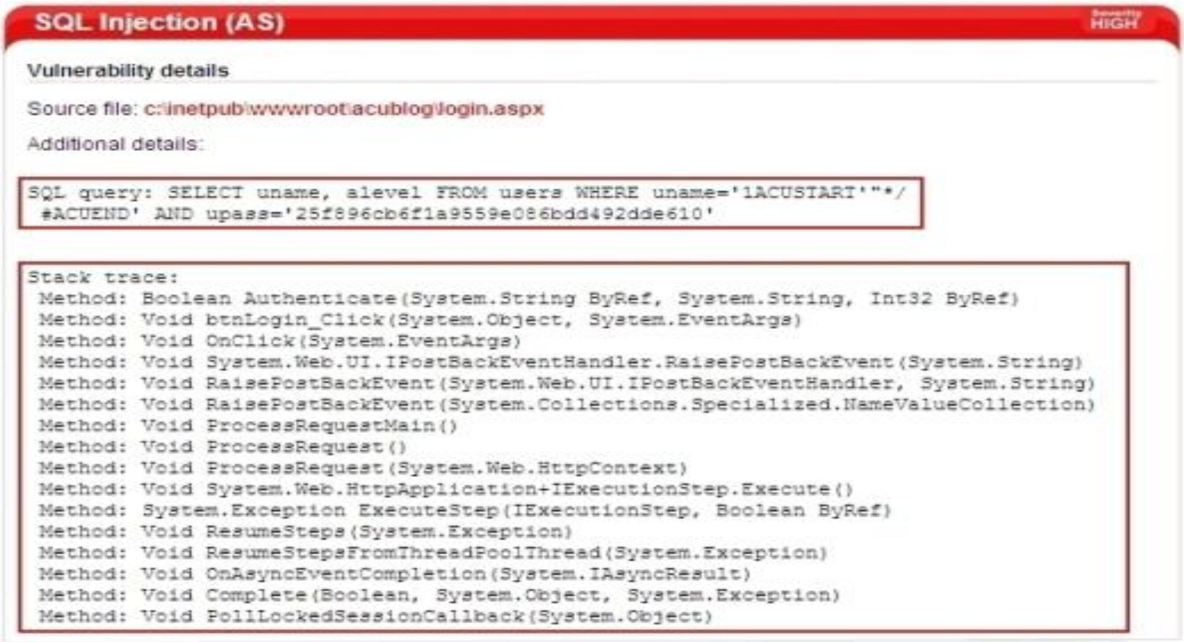
For this reason, Acunetix aims to crawl different versions of each website with different user agents.

SQL injection vulnerability can cause the exposure of all the sensitive data including username, password, and credit card details of a web application database (Khalid & Yousif, 2016). The SQL attacker tries to insert a part of malicious SQL commands by using special variables and inserting them in to the application. The web application in turn sends these malicious commands to the target database in the server that executes them in a different purpose using legitimate query (Abdulqader, Thiyab, Ali, & 2017). A blind SQL injection involves asking the database a series of true or false questions, in order to get closer to the vulnerable code itself. It is important to make a note here that identifying vulnerable code itself may not be sufficient for hackers on its own. Additionally, phishing is often used to obtain other user details (Erturk, 2012). These details can then be used as part of an SQL injection attack to extract unauthorized information from an online database.

Directory traversal (also known as path traversal) can be defined as an attack that “aims to access files and directories that are stored outside the web root folder” (OWASP, 2015). This vulnerability can exist in the web server or web application code. This allows the attacker to access parts of directories which are restricted, and to execute commands on the web server.

Technologies

Acunetix uses technologies like AcuSenor and AcuMonitor to achieve better scanning results. Acunetix AcuSensor Technology is a security technology that allows the identification of more vulnerabilities with less false positives. In addition, it indicates the exact location of the code where the vulnerability is and reports debug information. This technology combines black box scanning techniques and feedback from sensors placed inside the source code to achieve more accuracy. The screenshots below show the SQL injection (Figure 11) and PHP vulnerabilities (Figure 12) identified by the AcuSensor technology. It displays the stack trace of SQL injection threat and the file name with line number for the PHP code injection, which helps the developers to trace and fix vulnerabilities easily. It can also help developers to understand more about vulnerabilities which in turn helps them to write more secure code.



Vulnerability details

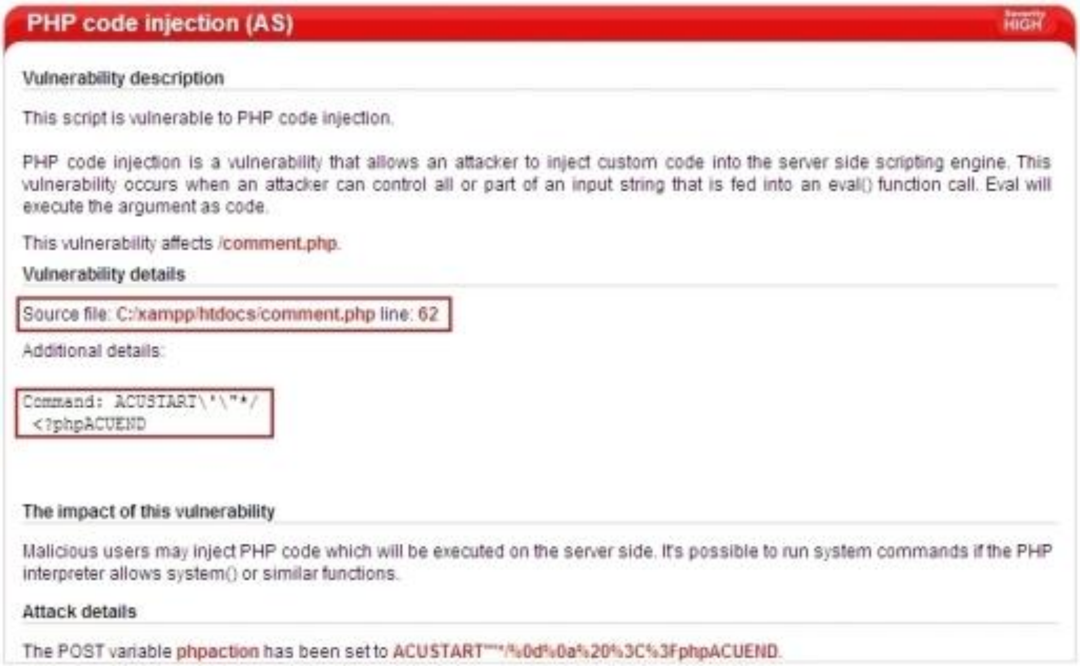
Source file: c:\inetpub\wwwroot\acublog\login.aspx

Additional details:

```
SQL query: SELECT uname, alevel FROM users WHERE uname='1ACUSTART'"*/
 #ACUEND' AND upass='25f896cb6f1a9559e086bdd492dde610'
```

```
Stack trace:
 Method: Boolean Authenticate(System.String ByRef, System.String, Int32 ByRef)
 Method: Void btnLogin_Click(System.Object, System.EventArgs)
 Method: Void OnClick(System.EventArgs)
 Method: Void System.Web.UI.IPostBackEventHandler.RaisePostBackEvent(System.String)
 Method: Void RaisePostBackEvent(System.Web.UI.IPostBackEventHandler, System.String)
 Method: Void RaisePostBackEvent(System.Collections.Specialized.NameValueCollection)
 Method: Void ProcessRequestMain()
 Method: Void ProcessRequest()
 Method: Void ProcessRequest(System.Web.HttpContext)
 Method: Void System.Web.HttpApplication+IExecutionStep.Execute()
 Method: System.Exception ExecuteStep(IExecutionStep, Boolean ByRef)
 Method: Void ResumeSteps(System.Exception)
 Method: Void ResumeStepsFromThreadPoolThread(System.Exception)
 Method: Void OnAsyncEventCompletion(System.IAsyncResult)
 Method: Void Complete(Boolean, System.Object, System.Exception)
 Method: Void PollLockedSessionCallback(System.Object)
```

Figure 11. SQL Injection Reported by AcuSensor Technology



Vulnerability description

This script is vulnerable to PHP code injection.

PHP code injection is a vulnerability that allows an attacker to inject custom code into the server side scripting engine. This vulnerability occurs when an attacker can control all or part of an input string that is fed into an eval() function call. Eval will execute the argument as code.

This vulnerability affects /comment.php.

Vulnerability details

Source file: C:/xampp/htdocs/comment.php line: 62

Additional details:

```
Command: ACUSTART\'\"*/
 <?phpACUEND
```

The impact of this vulnerability

Malicious users may inject PHP code which will be executed on the server side. It's possible to run system commands if the PHP interpreter allows system() or similar functions.

Attack details

The POST variable phpaction has been set to ACUSTART'"*/%0d%0a%20%3C%3FphpACUEND.

Figure 12. PHP Code Injection Reported by AcuSensor Technology

Another technology used by Acunetix is AcuMonitor Technology. While testing web applications, normally the scanner sends a request to a target, receives a response, analyses that response, and raises an alert based on the analysis. Some vulnerabilities do not give a response to a scanner during testing (Out-of-band vulnerability testing), and therefore, are not detectable using the "request/response" testing model. Detecting these vulnerabilities requires an intermediary service that a scanner can access. Acunetix, combined with AcuMonitor, makes automatic detection of such vulnerabilities easy. AcuMonitor detects Out-of-band SQL Injection (OOB SQLi), Blind XSS (or Delayed XSS), SMTP Header Injection, Blind Server-side XML/SOAP Injection, Out-of-band Remote Code Execution (OOB RCE), Host Header Attack, Server-side Request Forgery (SSRF), and XML External Entity Injection (XXE) automatically ("AcuMonitor," n.d.).

Discussion

Acunetix allows multiple scans simultaneously, but this may require more time for completing the entire scanning process. Time to scan a particular website or web application depends upon the technologies and complexity of the target website. Acunetix also allows to scan for particular vulnerabilities like XSS that can be set while starting of the scan. This is an advantage as this is much quicker compared to full scan and may help the developers to concentrate on a particular vulnerability and fix it. Reports generation is another good feature which helps to reuse the scan results. Different reports allow to get results based on specific need. Even though Acunetix reduces the false positives, scanning results may still contain false positives. Developers need to double check the scanning result and confirm that results are not false positives, which can be more time consuming. If Acunetix provides a feature to mark the false positives manually and restrict those from further scanning results, scanning results could have been better with less false positives.

Web Vulnerability Scanners (WVS) insert garbage values in the database while scanning (Suteva, Anastasov, & Mileva, 2014). WVS does automated scanning and performs several operations on a database to identify SQL injection and other threats related to the database. This can be an issue in the production environment as many garbage values are inserted with the original data.

An automated vulnerability scanner sends thousands of web requests to the web server. In order to accelerate their scanning, vulnerability scanners tend to send these requests using multiple simultaneous connections. If a web server is incapable of handling all the requests, the web server may slow down, resulting in a denial of service (Darmanin, 2014).

Automated WVS allows deep scanning. Thisi when a WVS tries to access all possible paths and links in website. This can be an issue while crawling into sensitive links. Crawling on sensitive links like delete can cause accidental deletion of some important data (Darmanin, 2014).

Some websites allow sending emails, for example, a “contact us” option in websites. While testing these websites, multiple emails can be sent to a particular address, as a mass mailing attack or mail flooding (Darmanin, 2014).

There are few solutions available for the problems discussed above. Testing in a staging environment, instead of production environment, can help prevent denial of service problems in the release environment (Darmanin, 2014). This also avoids insertion of garbage values in the production database. Denial of service in a production environment can be reduced. Using CAPTCHAs in the form for sending emails will help prevent email flooding. Furthermore, a WVS like Acunetix allows to restrict sensitive links from being crawled (Darmanin, 2014).

Mobile Device Aspects

Smartphones and tablets continue to pose some risks to users in terms of web security. These devices typically handle different wireless connections, cached or saved passwords, and notes and emails containing private information. As these bits of data are stored on mobile devices, they may be exposed to being stolen through web browsers. The education sector comprises the largest and growing groups of users of mobile devices. School teachers, employees and students constantly rely on their websites and online learning tools, and these are frequently facing security threats (Levin, 2017). More than ever, the users of educational information systems need to be more alert and better trained in security matters. IT support services for schools need to build their protective capacity and carry out new security practices. Mobile and web security aspects also need to be covered by analysts and developers of educational applications by making security a crucial requirement and design principle (Erturk, 2013).

Conclusions

Web Vulnerability Scanners (WVS) help to speed up the website and web application vulnerability testing process. Acunetix is a popular automated vulnerability scanner, which not only identifies vulnerabilities, but also gives suggestions for solving these vulnerabilities. AcuSensor technology also focuses on reducing the reporting of false positives for websites based on PHP and .NET technologies. utilizing new behavioural analysis techniques in the future can make Acunetix even more accurate. Using of Acunetix or a similar WVS is very useful for many web developers since this type of tool makes the detection of threats easier for security novices. Using a trial version can be helpful in make the decision regarding how long a period to choose for the paid subscription in the future. Different plans help the users to choose a suitable subscription option according to their needs. Website and application developers can use WVS during their development and testing to ensure a very secure web application before it is released into the production environment. This case study is helpful for orienting students with the basics of a WVS, with Acunetix as an example. Whereas some security solutions are geared towards examining organisational websites, other solutions may focus on managing the organisation's mobile devices. It is useful for both app designers and for IT support to scan mobile devices to identify the security vulnerabilities and to highlight insecure apps (Revankar, 2015). Further technical studies can be done to compare different vulnerability scanners, their effectiveness, and their peculiar strengths, which in turn would help developers choose an appropriate WVS for each web application.

References

Abdulqader, F. B., Thiyab, R. M., & Ali, A. M. (2017). The impact of SQL injection attacks on the security of databases. In Proceedings of the 6th International Conference on Computing and Informatics (pp. 323-331).

Acunetix - Website security - keep in check with Acunetix. (n.d.). Retrieved from https://www.acunetix.com

AcuMonitor: For detecting an XXE attack, Blind XSS and SSRF - Acunetix. (n.d.). Retrieved from https://www.acunetix.com/vulnerability-scanner/acumonitor-technology/

Cloud Security Alliance. (2016). The Treacherous 12: Cloud Computing Top Threats. Retrieved from https://downloads.Cloudsecurityalliance.org/assets/research/top-threats/Treacherous-12_Cloud-Computing_Top-Threats.pdf

Darmanin, G. (2014, May 5). Negative impacts of automated vulnerability scanners and how to prevent them. Retrieved from https://www.acunetix.com/blog/articles/negative-impacts-automated-vulnerability-scanners-prevent/

Erturk, E. (2012). Two Trends in Mobile Malware: Financial Motives and Transitioning from Static to Dynamic Analysis. Infonomics Society.

Erturk, E. (2013). An intelligent and object-oriented blueprint for a mobile learning institute information system. International Journal for Infonomics (IJI), 6(3/4), 736-743.

Gupta, S., & Gupta, B. B. (2015). PHP-sensor. *Proceedings of the 12th ACM International Conference on Computing Frontiers - CF '15*. doi:10.1145/2742854.2745719

InfoSec Institute. (2014, September 24). 14 best open source Web Application Vulnerability Scanners. Retrieved from http://resources.infosecinstitute.com/14-popular-web-application-vulnerability-scanners/#gref

Jasmine, M. S., Devi, K., & George, G. (2017). Detecting XSS based Web Application Vulnerabilities. *International Journal of Computer Technology & Applications*, *8*(2), 291-297.

Jeeva, S., Raveena, K., Sangeetha, K., & Vinothini, P. (2016). Web Vulnerability Scanner using Software Fault Injection Techniques. *International Journal of Advanced Research Trends in Engineering and Technology*, *3*(2), 637-649. Retrieved from https://www.researchgate.net/publication/303756552_WEB_VULNERABILITY_SCANNER_USING_SOFTWARE_FAULT_INJECTION_TECHNIQUES

Khalid, A., & Yousif, M. F. (2016). Dynamic analysis tool for detecting SQL injection. *International Journal of Computer Science and Information Security*, *14*(2), 224-232. Retrieved from https://www.researchgate.net/publication/311081330_Dynamic_Analysis_Tool_for_Detecting_SQL_Injection

Levin, D. (2017, March 14). How Should We Address the Cybersecurity Threats Facing K-12 Schools? Retrieved from https://www.edtechstrategies.com/blog/how-should-we-address-cybersecurity-threats-facing-k-12-schools/

Lyne, J. (2013, September 6). 30,000 Web Sites hacked a day. How do you host yours? Retrieved from https://www.forbes.com/sites/jameslyne/2013/09/06/30000-web-sites-hacked-a-day-how-do-you-host-yours/#4d3626541738

OWASP [Open Web Application Security Project]. (2015, October 6). Path Traversal. Retrieved from https://www.owasp.org/index.php/Path_Traversal

Revankar, M. (2015, October 15). Mobile Device App Inventory Auditing with Nessus 6.5. Retrieved from https://www.tenable.com/blog/mobile-device-app-inventory-auditing-with-nessus-65

Saeed, F. A. (2014). Using WASSEC to evaluate commercial Web Application Security Scanners. *International Journal of Soft Computing and Engineering*, *4*(1), 177-181. Retrieved from https://www.researchgate.net/profile/Fakhreldeen_Saeed2/publication/311310455_Using_WASSEC_to_Evaluate_Commercial_Web_Application_Security_Scanners/links/5879149c08ae9275d4d91b83/Using-WASSEC-to-Evaluate-Commercial-Web-Application-Security-Scanners.pdf

Schupak, A. (2015, March 24). One in three top websites at risk for hacking - CBS News. Retrieved from http://www.cbsnews.com/news/one-in-three-websites-at-risk-for-hacking/

Suteva, N., Anastasov, D., & Mileva, A. (2014, April). *One unwanted feature of many Web Vulnerability Scanners*. Paper presented at Proceedings of the 11th International Conference on Informatics and Information Technologies.

Warman, M. (2013, January 15). 90 per cent of passwords ‘Vulnerable to Hacking’- Business Insider. Retrieved from https://www.businessinsider.com.au/90-percent-of-passwords-vulnerable-to-hacking-2013-1?r=US&IR=T