Exploit Vulnerabilities of LAMP Based Web Applications in DETERlab

Jacob M. Hadden

Computer Science Texas A&M University - Corpus Christi jhadden@islander.tamucc.edu

Graduate Mentor: Jia Bai and Xiaowei Li Faculty Mentor: Prof. Yuan Xue

July 31, 2010

TRUST Research Experiences for Undergraduates (TRUST-REU) in Cyber Security and Trustworthy Systems 2010



Institute for Software Integrated Systems Vanderbilt University

Exploit Vulnerabilities of LAMP Based Web Applications in DETERlab

Jacob M. Hadden

Abstract

The DETER testbed is a secure infrastructure that is used to test cyber-security in a realistic environment. This paper presents three experiments involving web application vulnerabilities tested on DETERIab nodes. These three vulnerabilities are: SQL Injection, Command Injection, and File Inclusion.

1 Introduction

Web application account for 89% of all web related vulnerabilities contain vulnerabilities for are a way for attackers to infiltrate and gain control of web servers [1]. Web application vulnerabilities have become so prevelant that organizations such as The Open Web Application Security Project (OWASP), for the sole purpose of improving the security of application software [2]. DETERlab is a secure testbed that emulates medium sized networksand is set up in a way that mimics the Internet. This environment may be ideal for testing web application security methods. To validate this claim, and to determine if a DETERlab network would be a good teaching and training environment for web application security we conduct

2 Experiment Preperation

To test DETERIab's capabilities as a network emulator for cyber defense training and education we choose a three-node, non-cyclical topology in order to control the direction of traffic. The Security Experimentation EnviRonment (SEER) tool is used to create the chosen topology using Ubuntu 8.04 as the standard operating system on all three nodes. We swapin the DETERIab experiment with our chosen topology. After the experimental nodes are up and running, we install the necessary software for each node. The first node, Node LAMP (Linux, apache, mySql, and php), is set up according to a basic guide for a linux web server found on http://www.mysql-apachae-php.com. LAMP is installed with Damn Vulnerable Web Apps, DVWA, a vulnerable web application package made by Duncan Alderson forpenetration testing experience for experts and a learning experience for the inexperienced. The second node, Node Shark(Wireshark/tShark and Snort), was set up with an Intrusion Detection System(IDS) and packet sniffers. Shark's purpose will be to monitor the traffic, the important data for this experiment will be the packets recorded by wireshark. These packets are checked for interference from DETERIab. The packets may also be used for future works. The third node, Node Hack Lynx web browser, is installed

with exploit tools, the tools in use for these experiments are the Lynx web broswer for Linux. Lynx's command -logcommand is used to capture keystrokes for the three attacks chosen for this experiment so that a script may be used to automate the attacks.

3 Methods

The network setup has allowed for attacks against webapplications. After exploring the DVWA website hosted on LAMP we choose to attack the SQL Injection, Command Execution, and File Inclusion pages. Each page is attacked 30 times and packets are captured by Shark.

3.1 SQL Injection

We attack the SQL Injection page by insering the command 23 or 1=1 into the submission field. This give us all users and passwords in the data base, since the field is left vulnerable on the page. After we log the key strokes using Lynx we make a script and repeat the process 30 more times and have Shark record the traffic. This last step is followed for all three vulnerabilities.

Toola Ticib		
http://localhost:8080/dvwa/vu	nerabilities/sqli/?id=23+or+1%	o3D1&Submit=Submit#
d 🔝 Latest Headlines 🔻		
🚱 isi.deterlab.net - Experin	ne 🗱 间 Damn Vulnerable	Web App 🗱 🖶
	Damn	/ulnerable Web App
		OL Inication
Home	vulnerability: S	QL Injection
Instructions		
Setup	User ID:	
Brute Force	23 or 1=1	Submit
Command Execution	ID: 23 or 1=1	
CSRF	First name: admin	
File Inclusion	Surname: admin	
SQL Injection	ID: 23 or 1=1	
Upload	Surname: Brown	
XSS reflected	ID: 23 or 1=1	
XSS stored	First name: Hack	N
	Surname: Me	M3
DVWA Security	ID: 23 or 1=1	
PHP Info	First name: Pablo Surname: Picasso	
About	TD: 32 of 1-1	
	First name: bob	
Logout	Surname: smith	

Figure 1. SQL Injection

3.2 Command Execution

We look at the source file for Command Execution and see that this page has been made secure against the command and the &&; however pipelining is still available. We insert — netstat into the submission page hit enter and discover this vulnerability is exploitable and we are able to use linux commands through this submission file.

3.3 File Inclusion

We attack the File Inclusion page by insering the tag "../../../../../../../etc/passwd" at the end of the url. Because of the insecurity of the php on this particular page, we are able to view the passwd file stored on LAMP.

😵 isi.deterlab.net - Experime	. ×	Damr	Vulnera	ble Web App 🕱	4			
Command Execution	Ine	tstat			submit			
CSRF								
	Acti	ve Inte	rnet co	nnections (w/o	servers)			
File Inclusion	Prot	o Recv-	Q Send-	Q Local Address		Foreign Addre	ss State	
SQL Injection	tcp		0	0 localhost:388	39	localhost:165	05 ESTABLIS	SHED
Unional	tcp		0	0 bpc204.1s1.de	terlab:www	users.1s1.det	erla:56679 ESTABLI	SHED
opioad	tcp		0	0 localhost:165	500	localhost:388	08 ESTABLIS	SHED
XSS reflected	ten		0	0 localhost:165	20	localhort:300	09 EDIADLI	SHED
VSS stored	ten		0	0 localhost:105	30	localhost: 165	00 ESTABLI: 05 ECTADIT	SHED
x33 80160	tcn		0	A localhost:388	36	localhost:165	AS ESTABLT	SHED
7	tcp		A	A hnc284 isi de	torlahiwa	users isi det	orla:55142 TIME WA	IT
DVWA Security	tcn		0	0 bpc204.isi.de	ter1:40848	users.isi.det	erla:16505 ESTABLT	SHED
PHP Info	udp		0	0 bpc204.isi.de	terl:44808	boss.isi.dete	rlab.:8509 ESTABLI	SHED
	Acti	ve UNIX	domain	sockets (w/o se	ervers)			
About	Prot	o RefCr	t Flags	Type	State	I-Node	Path	
	unix	2	[]	DGRAM		6056	@/com/ubuntu/ups	tart
Logout	unix	2	[]	DGRAM		6294	@/org/kernel/ude	v/udevd
	unix	13	[]	DGRAM		11441	/dev/log	
	unix	3	[]	STREAM	CONNECTE	D 63212	/var/run/mysqld/	nysgld.sock
	unix	3	[]	STREAM	CONNECTE	D 63211		
	unix	2	[]	DGRAM		61415	8 8 80 8	
	unix	3		STREAM	CONNECTE	D 29465	/var/run/dovecot,	/login/default:
	unix	3		STREAM	CONNECTE	29464	100000000000000000000000000000000000000	
	unix	3		STREAM	CONNECTE	D 29463	/var/run/dovecot,	/login/default
	unix	2	11	STREAM	CONNECTE	0 29462	(upr/run/douocot	(login (dofault)
	unix	2	11	STREAM	CONNECTE	D 29460	/var/iu//dovecot,	/ cogrin/derautta
	unix	3	11	STREAM	CONNECTE	D 29459	/var/run/dovecot	/login/default
	unix	3	11	STREAM	CONNECTE	D 29457	/ war / san/ dovecor,	, cogran, derauer
	univ	2	1 1	CTDEAM	CONNECTO	0 20454	(upr/run/douocot	(login (dofnult)

Figure 2. Command Injection

< 🔶 • C 🛛 🏠 🗖	http://localhost:8080/dvwa/vu	Soogle	9,				
🖏 vanderbilt nashville 🗱	🛦 Grooveshark - Listen t 🗱	🚱 isi.deterlab.net - Page 🗱	Facebook Jacob Had 💲	🗱 📵 Damn Vulnerable Web 🗱	🕼 Understanding /etc/sh 🗱 📲	. v	
next = 0.0 nov/motion bindbash to ben = 00 nove/mot inheads darmor x = 11 aarmor / antidian / minkh in the 2.2 bits / minkh inheads y = 2.3 bits / minkh inheads areas / 2 bits / minkh inheads / minkh inheads / 2 bits / minkh inheads / minkh inheads / 2 bits / minkh inheads / minkh inheads / 2 bits / minkh inheads / minkh inh							
Warning: Cannot modify header information - headers already sent by (output started at /etc/passwd:12) in /var/wwwi/dwwa/dwwa/includes/dwwaPage.inc.php on line 244							
Warning: Cannot modify header inform							
		Damn Vul	Inerable Web App				

Figure 3. File Inclusion

4 Conclusion

The experiment was able to reveal some challenges in web application vulnerability testing, as well as simulating actual real world topographies

- 1. No more then 2 nodes on the same network.
- 2. The Ethernet links generated were generally out of range of acceptability for some sniffers. (ex. eth12, eth13, etc...).
- 3. Filtering traffic generated by DETERlab boss nodes in order to monitor the experimental nodes traffic only.
- 4. A fresh install of the images were required everytime the experiment was swapped in.

Challenge number one was not a problem for the tests ran in this experiment however, if an experiment attempting to test vulnerabilities of a network of nodes on the same IP subnet would. Challenge number two was easy to overcome but was still a reoccuring challenge in setting up this experiment. One proposed fix would be to allow nodes to have static ethernet connections in their creation file. Challenge number three only posed a problem during the assessment of traffic generated for each test run. After the IP address for the DETERlab boss nodes, nodes that monitor the DETERlab network, it was easy to filter out the traffic. The biggest challenge was number challenge number four. DETERlab allows users to save the state of a node as an image, which can then be used when creating experiments. This works well with software that does not use the MAC or IP address of the node it is working on. Which meant LAMP and SHARK had to be reconfigured every time the experiment was swapped in. To get around this challenge scripts were written to automate the installation process for each node. A proposed fix for this challenge in DETER would be to spoof the MAC and IP address and allow them to be statically set, or generate a permenant one upon the creation of each experiment. Based on the tests run, DETERlab is an acceptable and practical tool for teaching and training cyber defense. DETERlab allows for multiple boxes to be set up realatively quickly and allows a safe environment for testing vulnerabilities that may be to dangerous to test in an unsecure network. This would be ideal for universities or organization that are unable to support enough hardware and shelf space for a secure environment to run similar tests.

5 Future Work

Further work in this area would be to have a more exhaustive testing of web application vulnerabilities, not just to determine if they are possible on DETER, but also to determine if those vulnerabilities lead to vulnerabilities on the DETER system itself. More further work would include testing web application vulnerabilities on a website hosted on a node connected to a business network. This experiment would be better suited for cross network attacks based on a single nodes vulnerabilities.

6 Acknowledgements

This research was supported by funding from the Team Research for Ubiquitous Secure Technology Research Experience for Undergrads (TRUEST-REU) sponsored by the National Science Foundation (NSF). Hosted by Vanderbilt University and mentored by Dr. Yuan Xue, and Grad students Jia Bai,Xiaowei Li. Dr. Mario Garcia, Dr. Scott King, and Mr. Steve Alves, from Texas A&M University -Corpus Christi, for their support and encouragement.

References

- [1] M. Khera. Web application security trends report. Online Document, Q3-Q4 2009. http://www.cenzic.com/downloads/Cenzic_AppSecTrends_Q3-Q4-2009.pdf.
- [2] A. van der Stock David Lowery David Rook. Owasp code review guide, 2002-2008.