



5 Cloud and network attacks and reverse engineering

Table of Contents

5 Cloud and network attacks and reverse engineering.....	1
Report.....	2
5.0 Cloud intro and preparations/prerequisites (pre-lab).....	2
5.1 SANS – Cloud Scanning.....	3
5.1.1 Cloud Scanning.....	4
5.1.2 Cloud Bucket Discovery.....	4
5.1.3 Cloud Investigation (voluntary).....	6
5.2 SANS – Cloud Application Attacks.....	6
5.2.1 Microsoft 365 Password Attacks.....	7
5.2.2 Cloud SSRF/IMDS Attack.....	8
5.3 BHIS – BTC (Breaching The Cloud) – (voluntary).....	9
5.3.1 AzureGoat.....	9
5.3.2 PurpleCloud and other cloud training material.....	11
5.3.3 Cloud videos, guides and presentations.....	11
5.4 Wireshark – Attack and Reconnaissance Signatures.....	12
5.4.0 Introduction to the task.....	12
5.4.1 Task questions and help.....	13
5.5 Reverse engineer managed code.....	14
5.6 Analysis of an unknown binary file.....	14
5.7 Lab Feedback.....	15
Appendix 1 - Am I allowed to scan.....	16
Appendix 2 – Using Wireshark.....	19
Note! Voluntary and deprecated tasks below this point.....	20
5.10 Intro and WiFi prerequisites.....	21
5.10.1 Theoretical WiFi security questions.....	22
5.10.2 Questions regarding your lab environment.....	22
5.11 Crack WEP.....	23
5.12 Crack WPA/WPA2 – PSK (WiFi Protected Access).....	25
5.13 Challenge from bitsec.se.....	26
5.14 Breaking WPA(2)-PSK with pre-computed hash tables and brute force (GPGPU) software.....	27
5.14.1 Using Cowpatty with a pre-computed hash table.....	27
5.14.2 Using Pyrit with a wordlist.....	28
5.14.3 Using Pyrit with an attack db.....	28
5.14.4 Using Pyrit with GPGPU support (voluntary extra task).....	29

This lab is quite long but, it is needed to scratch on the surface to necessary cyber security techniques.



Needed equipment and software

- See the previous labs for the essential stuff (Linux, networking, virtual machines).
 - For VM:s this means setting up network, keyboard, language, time etc.
- Optional non-simulated cloud
 - Access to a cloud where you can create infrastructure. Preferable with 30 days free tier so you can have an infrastructure running during the lab.
 - An Azure, AWS or GCP tenancy/subscription is preferred for simplicity, since most Terraform hacking lab setup tools are available for these cloud vendors ATM.
- Most of the needed resources are found here
 - https://users.du.se/~hjo/cs/gdt2y3/cloud_hack/ > ...

Report

When you are finished take a screenshot of your screen with the solved lesson if it is needed. You have to show in some way that it is your computer/user the screenshot is taken from. Attach answer to questions and eventual comments in a document and hand in as a compressed file in Learn.

5.0 Cloud intro and preparations/prerequisites (pre-lab)

According to cybersecurity experts, the most pressing cloud security challenges are

- misconfiguration of the cloud infrastructure (68 percent);
- unauthorized access (58 percent);
- insecure API (52 percent);
- accounts, services or traffic hijacking (50 percent) and
- external data sharing (43 percent).

Fact checked source: <https://www.cloudwards.net/cyber-security-statistics/>

Begin with listen to a Swedish podcast from Infosäkpodden about cloud security

- #6 - Molnsäkerhet - 2022-11-21 (31 min):
<https://poddtoppen.se/podcast/1641450086/infosakpodden/infosakpodden-6-molnsakerhet>

The cloud lab part is mostly based on material from two free SANS resources that origins from the SANS SEC504 course: <https://www.sans.org/cyber-security-courses/hacker-techniques-incident-handling/> which usually cost around \$8000+ to attend!

- A late 2021 SANS Workshop webcast from Joshua Wright “Cloud Attacks and Incident Response”: <https://www.sans.org/webcasts/tech-tuesday-workshop-cloud-attacks-incident-response/>
- A partly (only scanning) updated 2023 SANS blog post from Joshua Wright “Cloud Scanning for Vulnerability Discovery” <https://www.sans.org/blog/cloud-scanning-vulnerability-discovery/> that explain that nmap may not be the right tool for the cloud.
 - Cloud Scanning For Vulnerability Discovery YouTube video (January 4, 2023):
<https://www.youtube.com/watch?v=Izy9APT5e7I>



- A 2023 SANS Workshop webcast from Joshua Wright “Cloud Application Attacks”:
<https://www.sans.org/webcasts/sans-workshop-cloud-application-attacks/>

The third resource should be listened to if the voluntary part in the lab is to be performed where you hack the cloud in a real environment.

- Material from BHIS: <https://www.blackhillsinfosec.com/> Breaching the Cloud (BTC) guides and cloud courses by Beau Bullock

5.1 SANS – Cloud Scanning

Introduction

Participate in the free SANS Workshop webcast “Cloud Attacks and Incident Response” and the more recent blog post “Cloud Scanning For Vulnerability Discovery” (original links are given in section 5.0):

- Local downloads: http://users.du.se/~hjo/cs/gdt2y3/cloud_hack/SANS/ >
 - Cloud.Attacks.and.Incident.Response
 - Prerequisites, prior to the workshop actions and system requirements are found in the local Cloud.Attacks.and.Incident.Response/**README-URL.txt** file.
 - Cloud.Scanning.for.Vulnerability.Discovery
 - The CloudScanningforVulnerabilityDiscovery.pdf file contains the same information as the SANS blog article (html) in section 5.0.

We may perform cloud scanning against non-simulated targets according to Joshua Wright. Listen to the “Cloud Scanning For Vulnerability Discovery” YouTube video at around **time 52.00 to 54.20** where he explains why it should be legal. Its however your own responsibility if you scan networks “in the wild”. See Appendix 1 for more information and rules about scanning in the cloud.

Some links to tools, scripts and commands are given in the local Cloud.Scanning.for.Vulnerability.Discovery/**cloud-scan-urls.txt** file that can help you with OCI (Azure, AWS and GCP are already covered in the webcast and blog article).

OCI (Oracle Cloud Infrastructure) example

<https://docs.oracle.com/en-us/iaas/Content/General/Concepts/addressranges.htm>

```
$ wget https://docs.oracle.com/iaas/tools/public_ip_ranges.json
```

```
$ jq < public_ip_ranges.json '.regions | .[] | .cidrs | .[].cidr' -r > oci.txt
```

Notes

- At least 50 minutes or more is waiting time in the webcast.
- Webcast slides in the **webcast-121065.pdf** file.
- If you register at SANS (free) and login you can watch the SANS Tech Tuesday Workshop – Cloud Attacks and Incident Response Webcast (press the Watch Now button).
- Download the Cloud Application Attacks workshop lab VM.
- Login with the **username sec504** and the **password sec504**.
- The labs.zip and wiki.zip files are extracted from the SlingShot VM.
- **When inside the VM, visit the VMs built-in website at: <http://localhost:9001/>**
- Remember there is no outside network unless you turn it on.



The work to perform are

- Cloud Scanning (5.1.1)
 - Bonus (If Time Permits or Homework)
 - Blog post “Cloud Scanning For Vulnerability Discovery” and YouTube video
- Cloud Bucket Discovery (5.1.2)
 - Bonus (If Time Permits or Homework)

5.1.1 Cloud Scanning

Tasks

1. Perform the lab walk-thru exercise as described in SANS Workshop – Cloud Attacks and Incident Response (and the updated blog post “Cloud Scanning for Vulnerability Discovery”).

2. Perform the Bonus (If Time Permits or Homework) tasks

Tasks, questions to answer and report

a) Calculate how many potential IPv4 addresses that potentially can be scanned in a region for your data center provider of choice somewhere in Europe. You got the **countips.py** script for help. Report how you got to that answer.

b) Scan a private RFC-1918 (as 10.0.0.0/16, 172.16.0.0/16 or 192.168.0.0/16) cloud or real /16 network with masscan for some ports that use TLS. For example HTTPS (443), RDP (3389) etc. and measure the time and result. Show the command and generated output.

Post notes

An automated scan solution is Scout Suite <https://github.com/nccgroup/ScoutSuite> which probably is best installed as a container: <https://www.google.com/search?q=scoutsuite+docker+hub&oq=ScoutSuite+docker>

Another one is Pacu, but it is only suited for AWS:
<https://github.com/RhinoSecurityLabs/pacu>

5.1.2 Cloud Bucket Discovery

Tasks

1. Perform the lab walk-thru exercise as described in SANS Workshop – Cloud Attacks and Incident Response.

2. Perform the Bonus (If Time Permits or Homework) tasks.

Post notes

If time permits download a tool into your Kali, make a wordlist and execute. Many other/alternatives exist. Below is the end point format and some tools for various clouds.



AWS

- Updated bucket finder: <https://github.com/phx/bucketfinder>
- More: <https://linuxsecurity.expert/security-tools/amazon-s3-bucket-scanners>
- <https://s3.amazonaws.com/{BUCKETNAME}>

GCP

- GCP Bucket Brute: <https://github.com/RhinoSecurityLabs/GCPBucketBrute>
- <https://www.googleapis.com/storage/v1/b/{BUCKETNAME}>

Azure

- Basic Blob Finder: <https://github.com/joswrlght/basicblobfinder>
- <https://{ACCOUNTNAME}.blob.core.windows.net/{CONTAINERNAME}>

OCI

- <https://blogs.oracle.com/developers/post/working-with-oracle-cloud-infrastructure-object-storage>
- <https://{NAMESPACE}.compat.objectstorage.{REGION}.oraclecloud.com/{BUCKETNAME}>
- The actual content returned from a public buckets listing follows the AWSListBucketResult XML standard.

Using your stored OCI credentials and the OCI CLI you can list all files in a bucket with
`oci os object list --bucket-name <your-bucket-name> --all`
according to the Command Line Reference:

https://docs.oracle.com/en-us/iaas/tools/oci-cli/3.22.4/oci_cli_docs/cmdref/os/object.html

CLI: <https://docs.oracle.com/en-us/iaas/Content/API/Concepts/cliconcepts.htm>

Tasks, questions to answer and report

a) Can you find a bucket discovery tool for Oracle Cloud Infrastructure (OCI)? **This means a script that anyone can run which do not need any user credentials for to discover files in public buckets.**

If not make your own PoC in Python with less than 15 code lines or update the Azure “Basic Blob Finder” script (see link above) which is very similar for how to scan OCI.

Only one private and one public bucket needs to be tested, i.e. no wordlist is needed. You do not need to reveal your own OCI name space and bucket name.

OCI hints (function below can be used with the Azure Basic Blob Finder script)

a python function that print objects in a AWS or OCI bucket

```
def print_object_storage(xmlstr):  
    '''https://www.datacamp.com/tutorial/python-xml-elementtree'''  
    tree = ET.ElementTree(ET.fromstring(xmlstr))  
    root = tree.getroot()  
    for elem in root.iter():  
        if(elem.tag.endswith('Key')):  
            print(f'Key: {elem.text}')        if(elem.tag.endswith('LastModified')):  
            print(f'LastModified: {elem.text}')        if(elem.tag.endswith('Size')):  
            print(f'Size: {elem.text}')    print('')
```



b) Is it possible to tell the difference between if a bucket exists and is private **OR** if the bucket does not exist at all?

5.1.3 Cloud Investigation (voluntary)

Tasks

1. Perform the lab walk-thru exercise as described in the SANS Workshop – Cloud Attacks and Incident Response.

Tasks, questions to answer and report

None, this is for your own knowledge.

Post notes

s3logparse.py - A Simple Parser for AWS S3 Logs: <https://github.com/joswr1ght/s3logparse>

5.2 SANS – Cloud Application Attacks

Introduction

Participate in the free SANS Workshop webcast “Cloud Application Attacks” (original links are given in section 5.0):

- Local download:
http://users.du.se/~hjo/cs/gdt2y3/cloud_hack/SANS/Cloud.Application.Attacks
 - Prerequisites, prior to the workshop actions and system requirements are found in the local **README-URL.txt** file.

Notes

- At least 30 minutes or more is waiting time in the webcast.
- Webcast slides in the **webcast-123885.pdf** file.
- If you register at SANS (free) and login you can watch the SANS Workshop – Cloud Application Attacks Webcast (press the Watch Now button).
- Download the Cloud Application Attacks workshop lab VM.
- Login with the **username sec504** and the **password sec504**.
- The labs.zip and wiki.zip files are extracted from the SlingShot VM.
- **When inside the VM, visit the VMs built-in website at: <http://localhost:9001/>**
- Remember there is no outside network unless you turn it on.

The labs to perform are

- Microsoft 365 Password Attacks (5.2.1)
 - Bonus (If Time Permits or Homework)
- Cloud SSRF/IMDS Attack (5.2.2)
 - Bonus (If Time Permits or Homework)



5.2.1 Microsoft 365 Password Attacks

Tasks

1. Perform the lab walk-thru exercise as described in SANS Workshop - Cloud Application Attacks.
2. Perform the Bonus (If Time Permits or Homework) task and find users with a password from the small wordlist. Mittens2022 and Falsimentis123 are answered in the lab already, but remember we may find new accounts when using all the new 400 generated accounts.

Wordlist

- Password123
- Lakers2020
- Dodgers2020
- Mittens2022
- Falsimentis123
- Summer2022
- Coffee2022

Problems

If you get locked out from the built-in “AWS API Gateway” when password spraying (there are probably too few containers for this many users in the cloud simulation) just delete the current Fireprox endpoint, stop and start the containers again and create a new Fireprox endpoint. Then continue attacking from where you got locked out.

Questions to answer and report

- a)** Find all users from the 400 username list with passwords you could recover and report it.
- b)** If a user has MFA enabled on the account, how could we potentially bypass this protection?

Hints

- How to Find MFA Bypasses in Conditional Access Policies:
<https://www.youtube.com/watch?v=SK1zgqaAZ2E>
 - Local: http://users.du.se/~hjo/cs/gdt2y3/cloud_hack/ > How to Find MFA Bypasses in Conditional Access Policies
- To Install PowerShell in Linux follow the instructions here:
<https://learn.microsoft.com/en-us/powershell/scripting/install/installing-powershell-on-linux>



5.2.2 Cloud SSRF/IMDS Attack

Introduction

LFI/RFI and Server-Side Request Forgery (SSRF) are basically the same vulnerability that you may have used in another DVWA already. Read a discussion about it here:

<https://security.stackexchange.com/questions/191458/what-is-the-difference-between-rfi-lfi-and-ssrf>

Tasks

1. Perform the lab walk-thru exercise as described in SANS Workshop - Cloud Application Attacks.

2. Perform the Bonus (If Time Permits or Homework) task.

Questions to answer and report

a) Explain what the 169.254.169.254 IP-address really is and what it is used for in the cloud?

b) How can you from an instance in your selected cloud (a real one) view the Instance Metadata Service (IMDS) credentials via curl?

c) Try the SSRF/IMDS attack on your selected cloud tenancy using the included (cloud_hack local download) PoC-script **ssrf.py**.

Reference: <https://alexanderhose.com/how-to-hack-aws-instances-with-the-metadata-service-enabled/>

Is it possible via the web browser to get some sensitive instance metadata when calling the script? If so what url, commands, code, etc. did you use in this case?

Hint: You are allowed to change the content of the **ssrf.py** script if you use IMDSv2.

d) How can we prevent the abuse of SSRF/IMDS attacks?

Hints and help

Note that there may be a different approach compared to the SANS lab module when doing the IMDS retrieve attack. This may depend on the host operating system as well.

- PortSwigger explanation of SSRF: <https://portswigger.net/web-security/ssrf>
- AWS: <https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/instancedata-data-retrieval.html>
- Azure: <https://learn.microsoft.com/en-us/azure/virtual-machines/windows/instance-metadata-service>
- OCI: <https://docs.oracle.com/en-us/iaas/Content/Compute/Tasks/gettingmetadata.htm>
- GCP: <https://cloud.google.com/compute/docs/metadata/querying-metadata>
- Payload All The Things: <https://github.com/swisskyrepo/PayloadsAllTheThings> > SSRF



5.3 BHIS – BTC (Breaching The Cloud) – (voluntary)

Introduction

You should **first** listen to the videos and reading presentations (**#0, #2, and #3**) in section 5.3.3 and then follow the **step by step guide #1** (Set up your virtual Cloud Lab).

After this step you got a GNU/Linux VM which is prepared for cloud attacks in a non-simulated environment. By now you should recognize most of the tools present in the VM. You can apt update everything in the VM so Azure CLI, Terraform etc. will have its latest version.

Additionally you can update Ubuntu to 22.04-lts or 22.10, see:

<https://www.debugpoint.com/upgrade-ubuntu-22-04-22-10/> and install VS Code:

<https://www.golinuxcloud.com/install-visual-studio-code-ubuntu-22/> so you can edit scripts easier.

The cloud attack content in the VMs **lab_scenarios** folder or the Damn Vulnerable Azure/AWS/GCP Infrastructure (DV*I) scenarios are voluntary to perform. It requires you to have a tenancy at Azure, AWS or GCP so you can run the Terraform Goat-scripts at: <https://github.com/ine-labs> which create the necessary infrastructure.

I only have “free” access to Azure ATM so I give a little help from what I have tested. Regarding the exact daily cost of having the DV*I:s up running is hard to know. Costs are updated 6 times a day in Azure, so after 4 hours you should be able to estimate the total cost. As this is written I believe it is about **3 SEK/day** for AzureGoat so it is quite cheap entertainment :)

Looking into the Terraform script and setting `vm_size` to “Standard_B1s” you can confirm that the VM cost around 0.1 SEK/h according to: <https://azureprice.net/?region=swedencentral¤cy=SEK>

Some knowledge of Terraform is essential so you can provision the infrastructure in a nearby region and also limit the access by IP-number so the DV*I *Goat lab vulnerable VM instance is not wide open to the internet.

When logged in to Azure with AZ CLI you may have more than one subscription (similar to tenancy in OCI) that is Enabled. Ensure it is the correct one that will be used. Check the “isDefault”: true member in the JSON output.

It can be modified with the az commands below (id = subscription). Logout and in for the changes to apply.

```
$ az account list
$ az account set --subscription <name or id>
$ az logout
$ az login
```

5.3.1 AzureGoat

These are my changes following the walk-thru guide #1 with:

<https://github.com/ine-labs/AzureGoat>. The other DV*I:s (AWS and GCP) have a similar scenario.

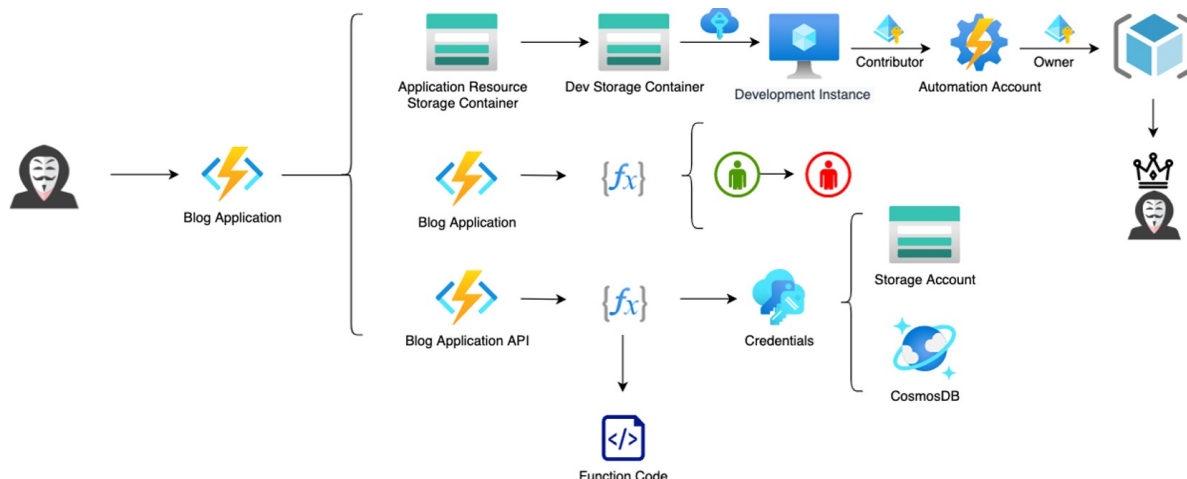


Figure 1. Overview of escalation paths for the AzureGoat module.

1. Create the azuregoat_app resource group closer to you as for example Sweden Central (swedencentral). Azure geographies: <https://azure.microsoft.com/en-us/explore/global-infrastructure/geographies/>

2. Before you run Terraform apply, edit **main.tf** so the region is provisioned correct. Edit the variable "location" to default = "swedencentral". By some reason there are some hard coded "location = eastus" strings which can be changed to "location = var.location". After this change everything will be provisioned to Sweden Central.

Also update the azurearm and random provider declarations to latest versions:

- <https://registry.terraform.io/providers/hashicorp/azurearm/latest>
- <https://registry.terraform.io/providers/hashicorp/random/latest>

Browse thru the code and verify that your IaC is OK, then enter terraform init, validate and plan. There are some warnings about deprecated code, but it is OK.

3. Perform terraform apply. During the deployment (which take some considerable time) you can watch the resource group filling/spinning up with various resources. Verify that they are created in your selected region.

Note the ending output

Target_URL = "https://appazgoat<random-number>-function-app.azurewebsites.net/"

4. After the deployment do not delete the local AzureGoat Terraform folder since we have the Terraform State there. Modify the Network Security Group so only your authorized IP:s can access AzureGoat via SSH. This can be done in the Terraform code as well before the deployment: https://registry.terraform.io/providers/hashicorp/azurearm/latest/docs/resources/network_security_group

5. Now you can try to hack the cloud application or cheat by following the solutions (text and video) at: <https://github.com/ine-labs/AzureGoat>. Part 04 and 05 is the most interesting IMHO and may be something you have not tried yet in the course.



6. When finished tear down all the created infrastructure with terraform destroy. That's why we need to keep the Terraform State intact so your computer knows exactly what to destroy. This also means that when creating additional stuff or modifying already created stuff, try to do it with Terraform so the state is updated.

Report

Nothing if you have not done anything. Otherwise I would like to hear about your experience.

5.3.2 PurpleCloud and other cloud training material

To perform the walk-thru guide #1 with: <https://github.com/iknowjason/PurpleCloud> and <https://www.purplecloud.network/install/#important-security-information-security-groups> you must have a domain in some way in Azure.

Here is a guide for how to do it for free: <https://nathancatania.com/posts/set-up-azure-active-directory-for-your-homelab/>

When this is done you can follow the guides exchanging the domain name with your domain name. This lab is however very similar to the "Microsoft 365 Password Attacks" lab dealing with Firefox and password spraying and so on.

The other labs not dealing with an AD as the storage lab should be OK to try without a domain.

Report

Nothing if you have not done anything. Otherwise I would like to hear about your experience.

5.3.3 Cloud videos, guides and presentations

#0. Breaching the Cloud Course Overview and introduction

- Breaching the Cloud Course Overview _ Beau Bullock:
<https://www.youtube.com/watch?v=aL3mGyLUmeI>
 - Local: http://users.du.se/~hjo/cs/gdt2y3/cloud_hack/BHIS-BTC/WebCasts/ > 0-Breaching the Cloud Course Overview _ Beau Bullock-aL3mGyLUmeI.mp4

#1. Set up your virtual Cloud Lab (with instructions)

- How to Build a Cloud Hacking Lab for Azure and more by Beau Bullock
<https://www.hubsite365.com/en-ww/crm-pages/how-to-build-a-cloud-hacking-lab-for-azure.htm> or
- How to Build a Cloud Hacking Lab - Beau Bullock https://www.youtube.com/watch?v=4s_3oNwqImo
 - Local http://users.du.se/~hjo/cs/gdt2y3/cloud_hack/BHIS-BTC/WebCasts/ > 1-How to Build a Cloud Hacking Lab-4s_3oNwqImo.mp4
 - Updated "How to Build a Cloud Hacking Lab" instructions

#2. Webcast Azure (lecture and presentation)



- EDITED EDITION - Getting Started in Pentesting The Cloud–Azure _ Beau Bullock
_ 1-Hour: https://www.youtube.com/watch?v=u_3cV0pzptY
 - Local: http://users.du.se/~hjo/cs/gdt2y3/cloud_hack/BHIS-BTC/WebCasts/ > 2-Getting Started in Pentesting The Cloud–Azure _ Beau Bullock-u_3cV0pzptY.mp4

#3. Webcast AWS (lecture and presentation)

- Securing AWS Discover Cloud Vulnerabilities via Pentesting Techniques _ Beau Bullock: https://www.youtube.com/watch?v=fg_hey18tio or
 - Local: http://users.du.se/~hjo/cs/gdt2y3/cloud_hack/BHIS-BTC/WebCasts/ > 3-Securing AWS Discover Cloud Vulnerabilities via Pentesting Techniques _ Beau Bullock-fg_hey18tio.mp4

#4. GitHub Cloud-Goats and tools

- INE Lab Infrastructure: <https://github.com/ine-labs>
- Rhino Security Labs: <https://github.com/RhinoSecurityLabs>
- Beau Bullock Dafthack -> CloudPentestCheatsheets etc.: <https://github.com/dafthack?tab=repositories>
- Black Hills Information Security: <https://github.com/orgs/blackhillsinfosec/repositories>
- Jason Ostrom: <https://github.com/iknowjason/>
- ...

#5. YouTube all (voluntary)

- Beau Bullock: https://www.youtube.com/@beau_bullock
- Black Hills Information Security: <https://www.youtube.com/@BlackHillsInformationSecurity>

5.4 Wireshark – Attack and Reconnaissance Signatures

5.4.0 Introduction to the task

Before doing this part of the lab it may be needed for you to refresh how the TCP Three-Way Handshake in TCP/IP connections and the ICMP protocol works.

More information:

- http://en.wikipedia.org/wiki/Transmission_Control_Protocol and
- http://en.wikipedia.org/wiki/Internet_Control_Message_Protocol
- PacketLife.net have handy study references (cheat sheets) for most things regarded to networking at: <http://packetlife.net/library/> including Wireshark display filters.
- A free chapter from the book "Wireless Sniffing with Wireshark" is available here: <http://users.du.se/~hjo/cs/common/books/Wireshark%20and%20Ethereal%20Packet%20Sniffing/>
- Another good refresh are the Wireshark instructor movies made by Jonathan James (<http://jonathanj.com/>) at:
 - [server]\networkanalysis or on YouTube
 - Kontrollera ditt nätverk med Wireshark – Dell: <http://www.youtube.com/watch?v=mgqUDSXiv08>



- Kontrollera ditt nätverk med Wireshark – Del2:

<http://www.youtube.com/watch?v=6uKYdgAyYiE>

- SecurityTube.net has some tutorial videos about Wireshark as well.
- NetworkMiner can be of great help as well: <https://www.netresec.com/>
- Laura Chappelle's WireShark lab (lots of packet challenges etc.): <https://www.chappell-university.com/lauras-lab>

5.4.1 Task questions and help

OS Fingerprinting - help for some of the question below

Ofir Arkin wrote XProbe2, a tool for determining the OS on a target system by sending a series of ICMP packets – type 13, 15 and 17, specifically:

- Type 13: Timestamp Request
- Type 15: Information Request
- Type 17: Address Mask Request

Most applications that use this method of OS fingerprinting put a signature into the communications. This signature resides in an ICMP Echo Request (type 8) packet which normally should have the Code 0. Some example of illegal ping packets:

- ICMP type 8 and Code 1 is NetScanTools (www.netscantools.com)
- ICMP type 8 and Code 123 is Xprobe2
- ICMP type 8 and Code 19 is LANguard

A great reference for signatures (or rules) is found at: www.snort.org. And most important thing to remember – **THE PACKETS NEVER LIE!**

Report

Use the attached pcap files in the pcap folder to answer the following questions.

- a)** Using the **osfingerprint.pcap** file make a Wireshark filter that displays all unique illegal ping packets.

Hint! Look in the trace file at packet 27.

- b)** Using **active-scan.pcap** and the illegal ping packet filter from task a. Try to find an ICMP echo request (type 8) packet with the wrong code value. What application (see list above) ran the scan?

- c)** Using **active-scan.pcap** note the number of TCP resets, how many?

- d)** Using **active-scan.pcap**, did the scanner look for any open UDP ports, if so which ones? Hint! Look for ICMP Destination unreachable (Port unreachable packets).

- e)** Using **sick-client.pcap** try to identify the bot information (user/nick, IP:port) the bot (bbjj.househot) use when connecting to its botnet (<http://en.wikipedia.org/wiki/Botnet>)?

- f)** This is a capture from a compromised system which boots up, runs for just about 3 minutes, then CPU utilization hits 100% and the system locks up. Using **clientdying.pcap** describe what is happening?



Hint! Start by looking at the protocols (Statistics > Protocol Hierarchy). We knew this client system didn't talk RPC, IRC or TFTP. Typically finding those applications scream hacked!

g) The **arp-poison.pcap** is a capture from a man-in-the-middle attack, see: http://en.wikipedia.org/wiki/Man-in-the-middle_attack. What IP and MAC address is the attacker having and who are the victims, explain for me please!?

5.5 Reverse engineer managed code

Report:

Describe the decoding/encoding algorithm in short and the found evidence for your statement.

Managed code is code like CIL (Common Intermediate Language) .NET and Java byte code.

Start the simplechallenge.exe program in the reverse-engineering\dot.net folder and try to figure out the decode/encoding algorithm. More information and hints are given in the programs dialog window and in the lecture presentation about obfuscation.

- Here is a guide for how to use Reflector: <https://www.simple-talk.com/dotnet/.net-tools/first-steps-with-.net-reflector/>
- The tool that replaces Reflector is ILSpy: <https://github.com/icsharpcode/ILSpy>
For Java and Android programs you can use JD-GUI: <http://java-decompiler.github.io/>
- To protect (obfuscate) your .NET programs use apps like: <https://yck1509.github.io/ConfuserEx/>

5.6 Analysis of an unknown binary file

Perform static and dynamic analysis of the file esh (unpack it with 7zip first). You can probably solve the whole problem via static analysis. Consult the presentations:

- EHP_14_malicious_code_eng.pdf
- EHP_15_malware_analysis_pen-test_methods_eng.pdf
to help you out.

Scenario

You work as an incident investigator and have got an unknown binary file sent to you via email from a colleague who wants your expert opinion about this file in a hurry.
Well well, another late night you think while you booting up the computers in your isolated malware analysis lab...

Report:

What are the true capabilities and functions of the file?

Document all the steps and commands you used and the motive for performing those in your answer. Attach relevant output and findings in your answer. You should be able to find the origins of the malware on the internet.



5.7 Lab Feedback

- a)** Was this a relevant and appropriate lab and what about length etc?
- b)** What corrections and/or improvements do you suggest for this lab?



Appendix 1 - Am I allowed to scan

First, two important notes:

- **Always get permission:** Check the Acceptable Use Policy for the cloud provider; you may need to get permission from the provider to perform the scan.
- **Scan within the provider and region:** When scanning, create the scanning system in the same cloud provider and region that you are scanning. If you are scanning multiple regions, then create multiple scanning systems. This will reduce cost and perform most optimally.

We may perform real cloud scanning in this part of the lab according to Joshua Wright. Listen to the “Cloud Scanning For Vulnerability Discovery” video at around **time 52.00 to 54.20** where he explains why it should be legal. Its however your own responsibility if you scan networks “in the wild” in this case.

Basically scans are allowed if one do not perform any testing for vulnerabilities in the category of “denial-of-service” and also to not disrupt any operations (customer or cloud vendor) activity.

It is very hard to detect if a scan is malicious if it is just one connection/packet to port 443. It can as well be a valid request.

Cloud vendors rules/policies about pen-testing

- Azure: <https://learn.microsoft.com/en-us/azure/security/fundamentals/pen-testing>
- AWS: <https://aws.amazon.com/security/penetration-testing/>
- OCI: https://docs.oracle.com/en-us/iaas/Content/Security/Concepts/security_testing_policy.htm
- GCP: <https://cloud.google.com/terms/aup>

The following info is a chat log from **ChatGPT**

Which cloud providers allows network scanning?

It is generally not recommended to perform network scanning on any cloud providers' infrastructure without explicit permission from the provider. Most providers have strict policies against unauthorized network scanning, and violations can result in account suspension or termination.

It is better to check with the specific cloud provider for information on their policies regarding network scanning and/or penetration testing.

Does Azure allow cloud network scanning?

Microsoft Azure, like most cloud providers, does not generally allow unauthorized network scanning on its infrastructure. Azure has strict security policies in place to protect customers' resources and data, and unauthorized network scanning is considered a violation of these policies.

However, Azure does provide a feature called Azure Security Center, which allows customers to perform vulnerability assessments and security scans on their resources, including network



scans. This feature is only available for customers who have enabled the Azure Security Center service and have the proper permissions.

Additionally, Azure also provides a feature called Azure Penetration Testing, which allows customers to conduct penetration testing on their resources with the approval of Azure. This service is intended to help customers identify and remediate vulnerabilities in their applications and infrastructure.

It is always recommended to check with Azure's documentation and compliance guidelines and also contact Azure support team before starting any kind of vulnerability testing or scanning activities.

Does AWS allow cloud network scanning?

Amazon Web Services (AWS) does not generally allow unauthorized network scanning on its infrastructure, as it is a violation of the AWS Acceptable Use Policy. This policy states that customers may not use AWS resources to engage in any activity that is illegal, malicious, or a violation of the security of any network, device, or data.

AWS does provide a feature called Amazon Inspector, which allows customers to perform security assessments and vulnerability scans on their resources, including network scans. This feature is intended to help customers identify and remediate vulnerabilities in their applications and infrastructure.

However, AWS customers are responsible for obtaining any necessary approvals or permissions before conducting any vulnerability testing or scanning activities on their resources. This includes any testing that may impact other customers, or that involves the use of shared resources, such as the AWS global infrastructure.

In addition to this, AWS also provides a feature called Penetration Testing with AWS, which allows customers to conduct penetration testing on their resources with the approval of AWS. This service is intended to help customers identify and remediate vulnerabilities in their applications and infrastructure.

It is always recommended to check with AWS's documentation and compliance guidelines and also contact AWS support team before starting any kind of vulnerability testing or scanning activities.

Does GCP allow cloud network scanning?

Google Cloud Platform (GCP) does not generally allow unauthorized network scanning on its infrastructure, as it is a violation of GCP's terms of service. This policy states that customers may not use GCP resources to engage in any activity that is illegal, malicious, or a violation of the security of any network, device, or data.

GCP does provide a feature called Cloud Security Scanner, which allows customers to perform security assessments and vulnerability scans on their resources, including network scans. This feature is intended to help customers identify and remediate vulnerabilities in their applications and infrastructure.

However, GCP customers are responsible for obtaining any necessary approvals or permissions before conducting any vulnerability testing or scanning activities on their resources. This includes any testing that may impact other customers, or that involves the use of shared resources, such as the GCP global infrastructure.

In addition to this, GCP also provides a feature called Penetration Testing, which allows customers to conduct penetration testing on their resources with the approval of GCP. This service is intended to help customers identify and remediate vulnerabilities in their applications and infrastructure.



It is always recommended to check with GCP's documentation and compliance guidelines and also contact GCP support team before starting any kind of vulnerability testing or scanning activities.

Does OCI allow cloud network scanning?

Oracle Cloud Infrastructure (OCI) does not generally allow unauthorized network scanning on its infrastructure, as it is a violation of OCI's terms of service. This policy states that customers may not use OCI resources to engage in any activity that is illegal, malicious, or a violation of the security of any network, device, or data.

OCI does provide a feature called Oracle Cloud Infrastructure Security, which allows customers to perform security assessments and vulnerability scans on their resources, including network scans. This feature is intended to help customers identify and remediate vulnerabilities in their applications and infrastructure.

However, OCI customers are responsible for obtaining any necessary approvals or permissions before conducting any vulnerability testing or scanning activities on their resources. This includes any testing that may impact other customers, or that involves the use of shared resources, such as the OCI global infrastructure.

In addition to this, OCI also provides a feature called Compliance and Security Assessment, which allows customers to conduct penetration testing on their resources with the approval of OCI. This service is intended to help customers identify and remediate vulnerabilities in their applications and infrastructure.

It is always recommended to check with OCI's documentation and compliance guidelines and also contact OCI support team before starting any kind of vulnerability testing or scanning activities.

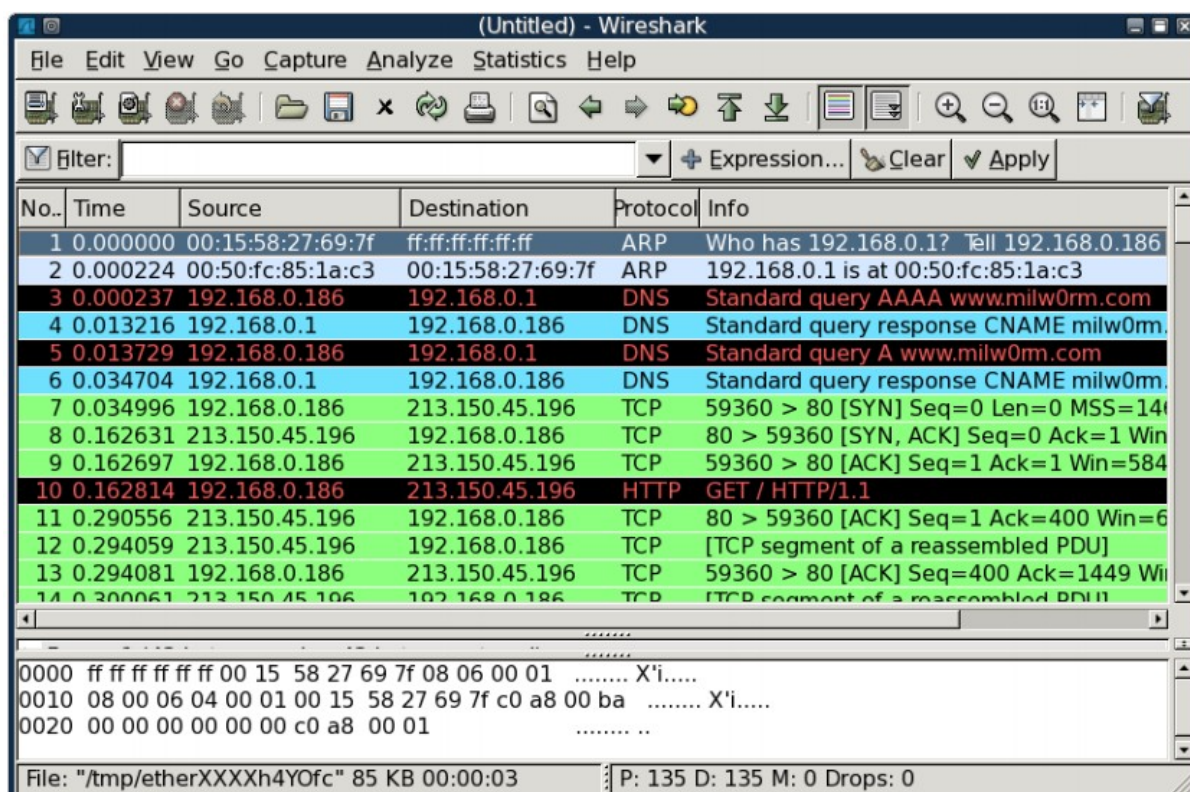


Appendix 2 – Using Wireshark

Learning how to use a sniffer effectively is probably one of the most important network related lessons one can take!

Let's begin by peeking into a Wireshark capture file. This capture was taken as I opened my browser and pointed it to <http://www.milw0rm.com> (as this is written the site is down, it contained a lot of exploits and vulnerable software).

However the site Inj3ct0r picked up where milw0rm left off at: <http://inj3ct0r.com/> and perhaps The Exploit Database to at: <http://www.exploit-db.com/>.



No.	Time	Source	Destination	Protocol	Info
1	0.000000	00:15:58:27:69:7f	ff:ff:ff:ff:ff:ff	ARP	Who has 192.168.0.1? Tell 192.168.0.186
2	0.000224	00:50:fc:85:1a:c3	00:15:58:27:69:7f	ARP	192.168.0.1 is at 00:50:fc:85:1a:c3
3	0.000237	192.168.0.186	192.168.0.1	DNS	Standard query AAAA www.milw0rm.com
4	0.013216	192.168.0.1	192.168.0.186	DNS	Standard query response CNAME milw0rm.
5	0.013729	192.168.0.186	192.168.0.1	DNS	Standard query A www.milw0rm.com
6	0.034704	192.168.0.1	192.168.0.186	DNS	Standard query response CNAME milw0rm.
7	0.034996	192.168.0.186	213.150.45.196	TCP	59360 > 80 [SYN] Seq=0 Len=0 MSS=1460
8	0.162631	213.150.45.196	192.168.0.186	TCP	80 > 59360 [SYN, ACK] Seq=0 Ack=1 Win=0
9	0.162697	192.168.0.186	213.150.45.196	TCP	59360 > 80 [ACK] Seq=1 Ack=1 Win=584
10	0.162814	192.168.0.186	213.150.45.196	HTTP	GET / HTTP/1.1
11	0.290556	213.150.45.196	192.168.0.186	TCP	80 > 59360 [ACK] Seq=1 Ack=400 Win=65535
12	0.294059	213.150.45.196	192.168.0.186	TCP	[TCP segment of a reassembled PDU]
13	0.294081	192.168.0.186	213.150.45.196	TCP	59360 > 80 [ACK] Seq=400 Ack=1449 Win=65535
14	0.300061	213.150.45.196	192.168.0.186	TCP	[TCP segment of a reassembled PDU]

File: "/tmp/etherXXXh4YOfc" 85 KB 00:00:03 P: 135 D: 135 M: 0 Drops: 0

Figure 2. Wireshark capture.

Looking at the screenshot above for the first time might be overwhelming. However we are going to examine the packet capture line by line and implement our knowledge in TCP/IP.

Packet 1: ARP Broadcast. We have attempted to send a packet to the Internet. Before our computer can actually send it, it needs to identify the default gateway on the local network. The default gateway IP address is configured on the requesting machine, but the default gateway MAC address is unknown. My machine sends a broadcast to the whole network, asking “Who has 192.168.0.1?, Tell 192.168.0.186”.

Packet 2: All computers on the local subnet receive this broadcast and check whether 192.168.0.1 belongs to them. Only 192.168.0.1 responds to this ARP broadcast and sends an ARP unicast reply to 192.168.0.186, informing it of the MAC address requested.



Packet 3: Now that our computer knows where to send its packets in order for them to reach the internet, we need to resolve the IP of www.milw0rm.com. Our computer sends a DNS query to the DNS server defined in our TCP/IP settings and asks the DNS server for the IP address of www.milw0rm.com.

Packet 4: The DNS server replies and tells our computer that the FQDN www.milw0rm.com is an alias for milw0rm.com.

Packet 5: Our computer asks the DNS server, once again, for the IP address of www.milw0rm.com. The first AAAA query was an IPv6 query and now we make an IPv4 query. Further info about this: http://en.wikipedia.org/wiki/List_of_DNS_record_types

Packet 6: The DNS server replies and tells our computer that the IP address for milw0rm.com is 213.150.45.196.

Packet 7: Armed with this information, our computer attempts a 3 way handshake with 213.150.45.196 and sends a SYN request (TCP flags = 0x02) on destination port 80. The source port is a high numbered dynamic port 59360.

- Well Known Ports (0 – 1023)
- Registered Ports (1024 – 49151)
- Dynamic and/or Private Ports (49152 – 65535)

Packet 8: The web server responds with an ACK which also contains a SYN (TCP flags = 0x12) on port 59360 to our machine.

Packet 9: We finally send an ACK to the web server and complete the 3 way handshake.

Packet 10: Now that the handshake is complete our computer can start talking with the service using a specific protocol. Since we are using a web browser, our computer sends an HTTP GET request which retrieves the index page and all linked images to our browser.

Packets 11 → end: ACK sent indicating clients previous package was OK. Now the main page at milw0rm.com, including all linked images, are going to be loaded in our browser. This will need a number of packets depending on the size of the HTML page.

Note! Voluntary and deprecated tasks below this point

Tasks below this point are not maintained and may not work any more.



5.10 Intro and WiFi prerequisites

This part is best done using the cloud as with most other attacks nowadays.

9 seconds to break a WiFi network using Cloud GPUs: <https://www.youtube.com/watch?v=nHDixd-EdEO>

Hundreds, perhaps thousands of articles have been written about the vulnerability of WEP (Wired Equivalent Privacy), but how many people can actually break WEP encryption? Beginners to WEP cracking have often been frustrated by the many wireless cards available and their distribution-specific commands. And things are further complicated when the beginner is not familiar with GNU/Linux since there is almost no equipment supporting wireless encryption attacks in Windows.

We are again going to use the Kali Linux or other environment preferred by you (no support is given by teachers in this case!).

First you should be familiar with your WiFi hardware, software and how the attack works. Therefore a good research should be done before starting this attack. In the WiFi references section I have put some of the (in my opinion) best information and instructions for this.

As earlier said there are countless resources for this subject on the web and the development of new attacks, software and hardware makes it hard to cover the absolute latest advancements. So if you have alternative or improved ways of doing the attacks it's a great idea using those!

Prerequisites for doing a full attack

- You must run Kali Linux in a non-virtual environment. For example Live via an USB drive: <https://www.kali.org/docs/usb/kali-linux-live-usb-install/>
 - You should also prepare for saving your capture files and lab report data to external storage as USB or network attached devices. There is a chance the captured data can be large and if you only store it on a RAM disk during the session it may be lost if the computer crashes.
- You must have a WiFi network card and driver which support RFMON (Radio Frequency MONitor) http://en.wikipedia.org/wiki/Monitor_mode in GNU/Linux. Some documentation about supported HW and SW: [http://www.aircrack-ng.org/doku.php?id=compatibility_drivers&s\[\]=supported&s\[\]=cards](http://www.aircrack-ng.org/doku.php?id=compatibility_drivers&s[]=supported&s[]=cards)
- You must have an environment similar to fig1.

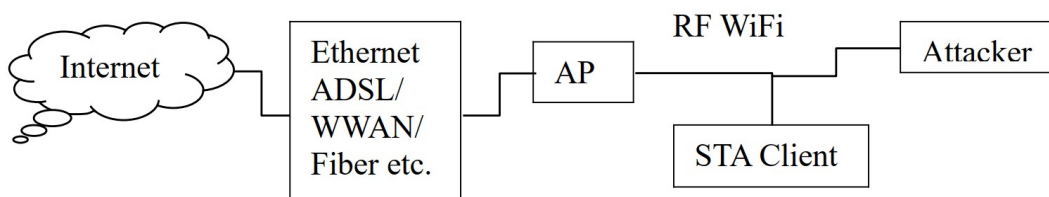


Figure 3. Attack components. AP with internet access, a STA (station) client (victim, computer, smartphone etc.) and an attack computer running GNU/Linux native.



Note! It may be possible to run Kali Linux in a virtual environment, but then you must have a RFMON capable USB-WiFi adapter which you can attach as a removable device in Vmware/VirtualBox.

At school we have the following USB based network adapters:

- 1 pc. DLINK DWA-140, chipset Ralink 2870
- 2 pc. DLINK DWA-172, chipset RTL8811au
- 4 pc. Linksys WUSB6400M AC1200, ??

Note! Managing the WiFi adapter in Kali Linux for normal wireless usage can be done with the following commands:

- iwconfig to see wireless adapters and then
- ifconfig <wlan#> up

There are many guides for connecting GNU/Linux to WiFi on the internet, for example:

https://www.youtube.com/results?search_query=Enable+WiFi+Kali+Linux

I have included a small **Wireshark guide** at the very end of this lab, “Using Wireshark”. I also recommend to have a look at: <http://wiki.wireshark.org/> especially the part about filters. Also be sure how to follow traffic streams and clear filters etc.

Fix various other Kali Linux problems: <https://www.kali.org/kali-linux-documentation/>

5.10.1 Theoretical WiFi security questions

Order all the WiFi access point settings/standards below according to how secure they are. Motivate your selection of order and combine options when possible and meaningful. Begin the order of settings/standards with lowest security first.

- WPA-RADIUS
- WPA2-PSK TKIP
- WEP – 128 bit
- Wireless SSID Broadcast ON/OFF
- WPA2-PSK AES
- WPA2-RADIUS
- Wireless MAC Filter ON/OFF
- WEP – 64 bit
- Encryption disabled
- WPA-PSK AES
- WPA-PSK TKIP

Here you can try out some router emulators: <https://www.snbforums.com/threads/router-ui-emulators.30552/>

5.10.2 Questions regarding your lab environment

a) Give all information about your AP (Access Point) as

- Product and model name
- Firmware version



- MAC address (BSSID)
- ESSID
- Supported encryption methods

b) Give all information about your victim (if possible) and attack STA (WiFi cards) as

- Product and model name of each
- MAC address of each
- WiFi chipset (<http://en.wikipedia.org/wiki/Chipset>) of each. Try search for “wifi adapter chipsets” if you are unsure. Sometimes the same as product and model name of the WiFi card

c) Is it possible to broadcast sniff WLAN network traffic on an **encrypted OR unencrypted** wireless network (802.11x) with some of the tools with the network card in default mode, promiscuous mode or via the monitor/rfmon mode? What applies to the various network settings in the AP? Fill in the '?' in the matrix below with explanations (not just yes or no). **Help** to solve this question: <http://wiki.wireshark.org/CaptureSetup/WLAN>

Possible to sniff broadcast traffic in?

NIC mode/AP mode	encrypted	unencrypted
default mode	?	?
promiscuous mode	?	?
monitor/rfmon mode	?	?

Table 1. WiFi network interface mode vs AP encryption mode.

5.11 Crack WEP

I recommend you to follow the SmallNetBuilder article “WEP Cracking...Reloaded” found in references (also included as PDF). Alternatively use the Simple WEP Crack Aircrack-ng tutorials and troubleshooting tips at: <http://www.aircrack-ng.org/doku.php?id=tutorial>

Note that when everything is setup and working (for the WEP attack) it can take some time to get the proper amount of IVs (Initialization Vectors), 5-10 minutes is normal. It may speedup if you disconnect and connect STA ↔ AP or generate some traffic.

If you do not have any supported hardware to perform this attack you should be able to answer the questions in a theoretical way by using the included capture file.

Resources

I have included a capture file (capturefile-01.cap) in the capdata folder which you can use if you do not have supported hardware or are not able to get it to work. Using the Aircrack-ng Windows GUI at: <http://www.aircrack-ng.org/> it can look like in fig 2.

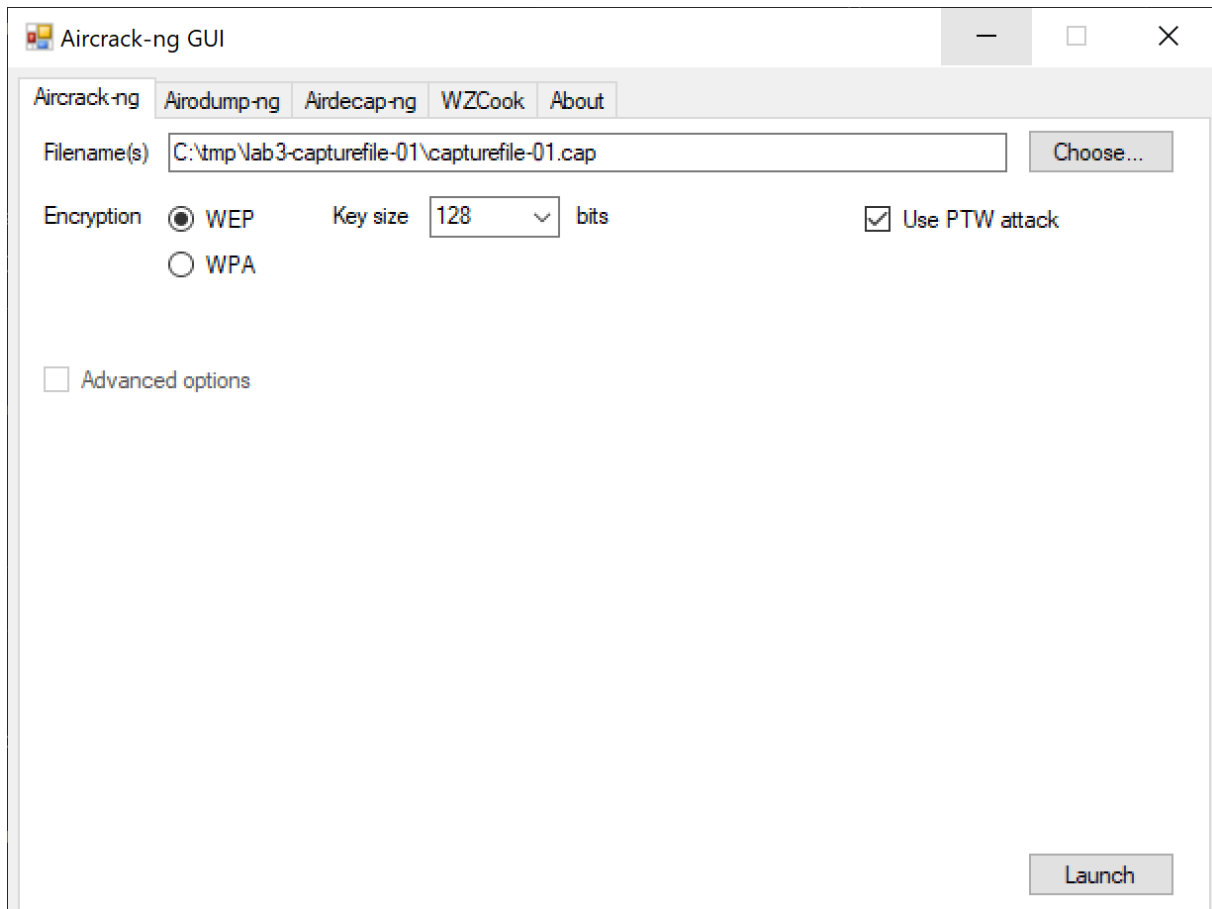


Figure 4. Aircrack-ng UI.

a)

- Try to find out the WEP key for the AP (Access Point) and note it down.
- Save the commands used and
- other important steps/output etc. in your attack.

b)

- Describe what method you used and
- how the attack actually works,
- what are the vulnerabilities?
- What prerequisites are needed to be met in order to make the attack etc.?

If needed show with a diagram or something else to make it simple to understand.

c)

When getting IVs using aircrack-ng and aireplay-ng it is slow in the beginning but speed up a lot when it actually starts, why does it behave like this?

d)

Attackers notes, what's your reflection as a pen-tester of this attack?



5.12 Crack WPA/WPA2 – PSK (WiFi Protected Access)

You basically do the same thing in this part as for cracking WEP but the attack is somewhat harder, different tools and more sophisticated techniques are needed. Before we can begin with this the AP must be reconfigured. So again this is the time to improve your theoretical knowledge about the attack.

I recommend you to follow the SmallNetBuilder article “How To Crack WPA/WPA2” found in references (also included as PDF). Alternatively use the How to Crack WPA/WPA2 tutorial and troubleshooting tips at: <http://www.aircrack-ng.org/doku.php?id=tutorial>

It is possible that you need to configure `/etc/kismet/kismet.conf` and add:
`ncsource=wlan0:type=madwifi`

Since the attack is targeted against the PSK (Pre Shared Key) the AP should have a password present in a wordlist/dictionary from one of the wordlists in Kali: `/usr/share/wordlists` or other wordlist/dictionary. Sometimes it can be really hard to get hold of the 4-way handshake as well.

If you do not have any supported hardware to perform this attack you should be able to answer the questions theoretically by using cowpatty.

Using Cowpatty with a pre-computed hash table

I have cowpatty pre-computed hash table files (genpkm) for two ESSIDs (linksys and test) at: `[server]\rainbowtables\wpa-psk\wpa_attack.7z`. They were downloaded from here: <https://www.renderlab.net/projects/WPA-tables/>

You can try coWPAtty for Windows (the .bat files in the folder `wpa_test`) with some of the included packet captures or your own acquired packet captures. You need a pre-computed hash table with matching ESSID (or a wordlist file, view the bat files above). Note that coWPAtty may need WinPcap: <http://www.winpcap.org/> to be installed in Windows to work.

In Kali Linux you can use something like this (check the cowpatty command options):
`# cowpatty -d linksys -r wpa(2)psk-linksys.dump -s linksys`

If you want to try against your own acquired packet captures you must have set your AP ESSID to a name present in a pre-computed hash chain table.

Valid PSKs (passwords) can be found in the `9-final-wordlist.zip` (`final-wordlist.txt`) which is used for these pre-computed tables. It is available here: `[server]\rainbowtables\wpa-psk\9-final-wordlist.zip`. For example the password “dictionary”, is used by one of the bat-file examples above.

a)

- Try to find out the WPA/WPA2-PSK password for the AP (Access Point) and note it down.
- Save the commands used and other important steps/output etc. in the attack.



b)

- Describe what method you used?
- How the attack actually works?
- What are the vulnerabilities?
- What prerequisites are needed to be met in order to make the attack etc?

If needed show with a diagram or something else to make it simple to understand.

c)

Attackers notes, what's your reflection as a pen-tester of this attack?

5.13 Challenge from bitsec.se

Report:

Du bör genomföra **minst punkt a** och ha en **riktig rapport** som stöd för dina antaganden. Alltså vilka ledtrådar den statiska och dynamiska analysen ger och ha en slags slutsats runt det. Lägg inte ner för mycket tid om du har mycket att göra.

Utförande och hjälp

Se filen bitsec_challenge.pdf för bakgrund och uppgift. Du kan välja själv hur långt du vill fullfölja. Om du sedan stannar vid fråga a, b eller utför hela uppgiften är upp till dig. Se tidigare nämnda: <http://old.honeynet.org/scans/scan32/> för ett bra exempel på rapporter.

Retargetable Decompiler är ett mycket bra onlineverktyg för reverse engineering. Det kan hjälpa dig att lösa delar av uppgiften. Se AVGs sida: <https://retdec.com/>



a) Undersök programmets binärkod som du lärt dig hittills i utbildningen.

b) Försök identifiera säkerhetsluckan.

c) Utveckla en exploit med root access.

För b och c behöver du troligen uppdatera dina kunskaper ifrån kursen Etisk hackning och penetrationstest. Kom ihåg att stänga av ASLR som root enligt bitsec instruktionen. Testa med:

```
// get_sp.c
#include <stdio.h>
unsigned long get_sp(void){
    __asm__("movl %esp, %eax");
}
int main(){
```



```
} printf("Stack pointer (ESP): 0x%x\n", get_sp());  
  
# gcc -o get_sp get_sp.c  
# ./get_sp  
Stack pointer (ESP): 0xbffffbd8 // remember that number for the next run.
```

5.14 Breaking WPA(2)-PSK with pre-computed hash tables and brute force (GPGPU) software

Report

Try cracking the password used in a WPA(2)-PSK key-negotiation with both brute force and a pre-computed hash table. Detailed instructions are given in sections 3.4.1 - 3.4.3.

a) Measure the difference. The number of passphrases/PMKs per second between cowpatty using a hash table.
pyrit using brute force with a wordlist.
pyrit using an attack db.

b) How fast is your hardware (CPU or GPU) with pre-computed tables compared to a GPU doing it all with brute force (Pyrit homepage or other resources may have performance numbers for different platforms)?

5.14.1 Using Cowpatty with a pre-computed hash table

I have put a number of cowpatty pre-computed hash table files (genpkm) for various ESSIDs on [server]\rainbowtables\wpa-psk\wpa_tables. They were downloaded from here: <https://www.renderlab.net/projects/WPA-tables/>

You can try cowpatty (the .bat files) in the attached folder wpa_test with some of the included packet captures or your own acquired packet captures. You need a pre-computed hash table with matching ESSID (or a wordlist file, view the bat files above). Note that cowpatty needs WinPcap: <http://www.winpcap.org/> to be installed on Windows to work.

In Kali Linux you can use something like this (check the cowpatty command options):
cowpatty -d linksys -r wpa(2)psk-linksys.dump -s linksys

A version of Pyrit may already be installed in Kali Linux. If not install it.

If you want to try against your own acquired packet captures you must have set your AP ESSID to a name present in the pre-computed hash chain tables. For example the **linksys** table name which is located here: [server]\rainbowtables\wpa-psk\wpa_tables\xai-0.

Valid PSKs (passwords) can be found in the included wordlist **9-final-wordlist.zip** (final-wordlist.txt) used for these tables. Also available here: [server]\rainbowtables\wpa-psk\



www.renderlab.net/wordlists. For example the password “dictionary”, which the example above use.

5.14.2 Using Pyrit with a wordlist

Text below is from the old Pyrit Wiki tutorial at:

<https://code.google.com/archive/p/pyrit/wikis>. The project is however now exported to GitHub: <https://github.com/JPaulMora/Pyrit/wiki> which contain updated info.

As you may already know, guessing the password used in a WPA(2)-PSK key-negotiation is a computational-intensive task. During this process, more than 99.9% of the CPU-cycles have to be spent to compute what is known as the *Pairwise Master Key*, a 256-bit key derived from the ESSID and a password using the PBKDF2-HMAC-SHA1-algorithm. One of the major weaknesses of WPA(2)-PSK is that the *Pairwise Master Key* has no elements that are unique to the moment of the key-negotiation between the AccessPoint and the Station. It is therefore possible to pre-compute the *Pairwise Master Key* and store it for later use.

In the moment of attacking a key-negotiation, we are left with the remaining 0.1% of what depends on session-unique data. It is therefore extremely valuable for an attacker to pre-compute large tables of *Pairwise Master Keys* for common ESSID:s.

Read thru the “First steps and tutorial” from the Pyrit Wiki tutorial link above. A version of Pyrit may already installed in Kali Linux. If not install it or perform the lab with one of the alternatives: <https://alternativeto.net/software/pyrit/>

In Kali Linux you can use something like this (check the pyrit command options):
`# pyrit -r wpa(2)psk-linksyst.dump analyze`

With information from the analyze you can perform the attack:

```
# pyrit -r wpapsk-linksyst.dump -i dictionary_wordlist.txt -b  
00:0b:86:c2:a4:85 attack_passthrough
```

5.14.3 Using Pyrit with an attack db

Using the database functionality in Pyrit is described in the Pyrit Wiki usage tutorial: <https://github.com/JPaulMora/Pyrit/wiki/Usage>. By creating an attack db you can effectively build your own pre-computed hash table for any ESSID.

In short you perform these commands to create an attack db/table for the ESSID linksys with the attached final-wordlist.txt.

```
# pyrit eval  
# pyrit -i final-wordlist.txt import_passwords  
# pyrit -e linksys create_essid  
# pyrit eval  
# pyrit batch
```



```
# pyrit -r wpapsk-linksyst.dump attack_db
```

5.14.4 Using Pyrit with GPGPU support (voluntary extra task)

If you got the suitable hardware, time over and want to experiment. **Note installing extra graphic drivers may break your GNU/Linux installation!**

Install Pyrit with GPGPU support according to some current internet resource.

You can search for “pyrit install openCL kali” or “pyrit install cuda kali” depending on your hardware.

Other ways to do fast cracking of WPA(2)-PSK passwords.

[https://www.google.com/search?](https://www.google.com/search?q=cloud+crack+password+wpa&oq=cloud+crack+password+wpa)

[q=cloud+crack+password+wpa&oq=cloud+crack+password+wpa](https://www.google.com/search?q=cloud+crack+password+wpa&oq=cloud+crack+password+wpa)

More information about cracking passwords in the article “Why passwords have never been weaker—and crackers have never been stronger” at:

<http://arstechnica.com/security/2012/08/passwords-under-assault/>