

# Practical Network Defense - Lab 9

Vulnerability assessment of ACME co.'s network

Alessandro Serpi - 1647244

17 May 2019

## Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>Greenbone setup</b>	<b>2</b>
2.1	Greenbone Community Edition in the local environment . . . . .	2
2.2	Tasks . . . . .	2
<b>3</b>	<b>Rules of engagement</b>	<b>3</b>
<b>4</b>	<b>Assessment results and analysis</b>	<b>3</b>
4.1	Cleartext transmission of sensitive information via HTTP . . . . .	3
4.2	Missing cookie attributes in the domain controller . . . . .	4
4.3	SSH brute force login with default credentials . . . . .	4
4.4	TCP timestamps . . . . .	4
<b>5</b>	<b>Mitigations</b>	<b>4</b>
5.1	Cleartext transmission of sensitive information via HTTP . . . . .	4
5.2	SSH brute force login with default credentials . . . . .	6
<b>6</b>	<b>Final remarks</b>	<b>6</b>

# 1 Introduction

Vulnerability scanners are applicative designed to discover weaknesses in an environment, especially those caused by misconfigurations or flawed software components. Continuous vulnerability scanning is a staple of an effective cyberdefense policy.

In this assignment we will execute a vulnerability scan of ACME co.'s private network. Then, we will identify and apply fixes to the discovered weaknesses. If it will be not possible to completely resolve an issue, we will at least try to mitigate the vulnerability.

## 2 Greenbone setup

### 2.1 Greenbone Community Edition in the local environment

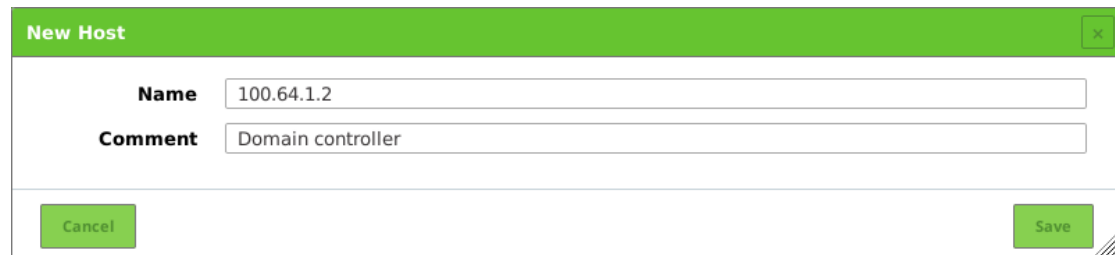
Create a new VirtualBox virtual machine *pnd-gsm*, selecting *Linux* and *Other Linux (64-bit)* as respectively OS type and version and assigning at least 4GiB of RAM and 9GiB of hard disk space. Then, download the latest ISO image of the Greenbone Community Edition and assign it to the CD interface of the virtual machine.

Start the virtual machine with the attached configuration script `start.vms.sh` using the parameter `gsm` and follow on-screen instructions to create a local and a web account.

Since the firewalls must allow all traffic from and to the vulnerability scanner, we decided not to place the virtual machine in the *DMZ* in order to not create holes in the firewall. Instead, we established it in the *INTERNAL\_SERVERS* network, where it is more difficult for a potential attacker to exploit the firewalls' configuration.

### 2.2 Tasks

Login in the web GUI and create a new host in *Assets* > *Hosts* for every machine in the network, filling *Name* with the host's IP address and *Comment* with a description.



Next, create new credentials in *Configuration* > *Credentials* for every used SSH user-name/password combination.

**New Credential**

**Name** Domain controller

**Comment**

**Type** Username + Password ▼

**Allow insecure use** ☐ Yes ☒ No

**Auto-generate** ☐ Yes ☒ No

**Username** earendil

**Password** \*\*\*\*\*

Cancel Save

In the host menu, create for each host a new target using the specific button. In the configuration window, select *All IANA assigned TCP and UDP 2012-02-20* as *Port List* and add the SSH credentials set (if it exists).

Finally, in *Scans* > *Tasks*, create a new task for every target, selecting *Full and fast ultimate* as *Scan Config* and *Random* as *Order for target hosts*.

### 3 Rules of engagement

We scanned all IANA-assigned TCP and UDP ports of every host in the private network. In addition, we gave to the vulnerability scanner SSH access to the resources in order to check for outdated and vulnerable components.

Clients, log server and web server were analysed only in the remote environment, while (due to technical problems) the domain controller and the coffee maker were checked only in the local environment. Consequently, the mitigation described in section 5.1 was not implemented in the remote environment.

## 4 Assessment results and analysis

### 4.1 Cleartext transmission of sensitive information via HTTP

*The host / application transmits sensitive information (username, passwords) in cleartext via HTTP. [...] An attacker could use this situation to compromise or eavesdrop on the HTTP communication between the client and the server using a man-in-the-middle attack to get access to sensitive data like usernames or passwords.*

In order to allow ordinary and extraordinary maintenance, the coffee machine's web GUI must be accessible from the outside. Since communications between users and vending machine are carried out in cleartext using HTTP, an attacker that has access to the

network can sniff user credentials or gain access to private data (such as the machine's current status).

## 4.2 Missing cookie attributes in the domain controller

*The application is missing the 'httpOnly' attribute. [...] This allows a cookie to be accessed by JavaScript which could lead to session hijacking attacks.*

*The flaw is due to cookie is not using 'secure' attribute, which allows cookie to be passed to the server by the client over non-secure channels (http) and allows attacker to conduct session hijacking attacks.*

The domain controller does not set 'httpOnly' and 'secure' cookie attributes, which can allow session hijacking attacks. Since this is a Zentyal's shortcoming, we can not implement any mitigation.

## 4.3 SSH brute force login with default credentials

*It was possible to login into the remote SSH server using default credentials. [...] Change the password as soon as possible.*

In the remote environment, we left the default password for the majority of the resources. There were no such vulnerabilities in the local environment: when we created the servers, we chose non-default username/password combinations.

## 4.4 TCP timestamps

*It was detected that the host implements RFC1323. [...] A side effect of this feature is that the uptime of the remote host can sometimes be computed.*

An attacker may use TCP timestamps to determine whether security patches requiring a reboot were applied to a host. However, disabling timestamps adds security through obscurity, which is no security. In addition, TCP timestamps are used in the PAWS (Protect Against Wrapped Sequence Numbers) mechanism.

Therefore, we decided not to consider the user of this functionality a vulnerability and not to implement any mitigation.

# 5 Mitigations

## 5.1 Cleartext transmission of sensitive information via HTTP

Since the vending machine does not support encryption, it is necessary to set up a TLS offloading mechanism to protect communications between the vending machine and the support crew. TLS operations are delegated to the main router, which encrypts messages coming from and decrypts messages directed to the vending machine.

Install *os-nginx* in *System* ▷ *Firmware* ▷ *Plugins*. Then, enable the web server in *Services* ▷ *Nginx* ▷ *Configuration*. From now on all actions will be performed in *nginx*'s configuration section.

Create a new upstream server for the vending machine in *Upstream* ▷ *Upstream Server* and configure it like in the following image (leave all other fields unchanged), ensuring that *Allow DNS server list to be overridden by DHCP/PPP on WAN* in *System* ▷ *Settings* ▷ *General* is disabled.

<b>Description</b>	Fantastic Coffee Maker
<b>Server</b>	100.64.4.10
<b>Port</b>	80

Create a new upstream in *Upstream* ▷ *Upstream* and configure it like in the following picture (leave all other fields unchanged).

<b>Description</b>	Fantastic Coffee Maker
<b>Server Entries</b>	Fantastic Coffee Maker ▼
	✖ Clear All

Create a new location in *HTTP(S)* ▷ *Location* and configure it like in the following picture (leave all other fields unchanged).

<b>Description</b>	Fantastic Coffee Maker
<b>URL Pattern</b>	/
<b>Match Type</b>	none ▼

Create a new certificate authority for the domain and a new server certificate for the subdomain `coffee-maker.pnndeflab.edu`.

Finally, create a new HTTP server in *HTTP(S)* ▷ *HTTP server* and configure it like in the following pictures (leave all other fields unchanged). It is especially important to select *HTTPS Only*, otherwise users would still be able to carry on insecure communications with the server machine.

HTTP Listen Port	<input type="text" value="80"/>
HTTPS Listen Port	<input type="text" value="443"/>
Server Name	<div>coffee-maker.pndeflab.edu ×</div> <div>✖ Clear All</div>
Locations	<div>Fantastic Coffee Maker ▼</div> <div>✖ Clear All</div>
TLS Certificate	<div>Fantastic Coffee Maker ▼</div>
CA Certificate	<div>PNDefLab ▼</div>
HTTPS Only	<input type="checkbox"/>

Even if ACME's web server does not receive or transmit sensible information, we nonetheless decided to secure it in the same way performing analogous steps.

## 5.2 SSH brute force login with default credentials

Use non-predictable passwords also in the remote environment.

## 6 Final remarks

Overall, we found no serious security holes in the local environment directly ascribable to our actions (or inaction): all machines were properly configured and up to date. On the other hand, in the remote environment we deliberately chose not to change the default passwords in order to ease third-party checks, diminishing the security of the network.