How to Circumvent a Captive Portal

Pierre Anderberg Erik Thorselius
Information Technology Programme
Department of Computer and Information Science
Linköping University
SE-581 83 LINKÖPING, SWEDEN
Email: {piean302, erith373}@student.liu.se

Abstract

The authorization mechanism in many wireless networks is to have a captive portal protecting the network from unauthorized users and for auditing the authorized users. In this paper we will describe how captive portals work and how they can be circumvented by relatively scarce resources, e.g via a laptop computer equipped with a standard network card and some open source software.

The conclusion from our experiments is that circumventing captive portals is easier said than done. According to non-mainstream literature from the hacker community captive portals offers poor protection for public networks but our experiments have shown that circumvention is possible but that the real benefits are low. The main reasons why benefits are low are as follows: in order for the attacker to be able to use the network in a stable fashion the victim needs to shut down his network when the attack is conducted, otherwise both the attacker and the victim will experience a unstable network and possibly lose connection temporary; and the attack can be thwarted easily if the victim signs out from the captive portal page.

1. Introduction

Public wireless networks are common and you can often find them in coffee shops, in schools, on trains or in other public environments. However, in many cases the owners of these networks wants to control the access to the networks for different reasons. Reasons could be that the network is only meant to be used by the employees or that the owner wants to collect some sort of fee from the users before they get access to the network. One way of achieving this control and authenticate the users is to use a captive portal.

A captive portal is basically a system that holds a unauthenticated user captive until some sort of web based authentication mechanism authenticates the user and in this paper we will describe how a captive portal works and how it can be circumvented with fairly easy means.

1.1. Questions at Issue

a. How does a captive portal work?
b. Where are the vulnerabilities in a captive portal and where is the most suitable entry point for a attack?
c. What equipment is needed in order to conduct a attack?

2. Method

In this paper we have chosen to use experiments as a method for data gathering and the goal of this project has been to develop our own software and conduct a real attack against a captive portal. However, our experimental method is combined with a classical literature study where our goal is to attain knowledge on how a captive portal is constructed, its functions, as well as what kind of possible attacks that there are.

2.1. The Literature Study

Since computer security and its illegal counterpart cracking are areas that are evolving very quickly it is hard to find up-to-date information in the mainstream literature. Considering this we have chosen to include non mainstream literature such as Internet based forums and mailinglists from both computer security professionals as well from so called hackers. One might argue that the reliability of these sources is low, but despite this there is an academic consensus that these sources are needed in order to understand the context of network security from both the attacker as well as the defender perspective.

2.2. The Experiments

Our experiments were conducted as follows: first we collected information about our attack target, that is a

specific captive portal, in this case the captive portal of Linköping university network. Secondly we examined the structure of the target and eavesdropped on its network traffic. We did also some basic testing in this step in order to get input for the next step. The third step was planned to be programming our on software for automated attacks but because of problems that occured in the first two steps we had to downsize our project to writing a script that took target network and channel and the target computer MAC ad IP address as parameters. The last step consisted of a series of tests against an isolated network that was similar to the real target. A fifth step would have been to conduct a attack against the real target but this we did not do because of legal issues.

In the second step we found that if the victim was connected and active our connection was flaky and unreliable. In order to prevent this problem we tried a technique commonly used for wireless key cracking. This techniqie disassocated the victim from the access point with special crafted and spoofed arp packages and the method worked fine to make the victim disconnect but most operating systems default behaviour is to reconnect again, so in practice this technique was quite useless for us.

3. Description of a Captive Portal

In this section we describe the infrastructure of a captive portal and how a captive portal is implemented.

3.1. Definition of a Captive Portal

In this paper we will use the following definition of a captive portal: a captive portal is a system that holds the unauthenticated user captive until some sort of web based authentication mechanism authenticates the user. The system will intercept all requests and force the user to an authentication website, and after a successful authentication process the system changes the gateway so that the user is no longer a captive.

3.2. The Infrastructure of a Network With a Captive Portal

A captive portal is a web page that the user of a public-access network is obliged to view and interact with before access is granted [**?**]. However, the term captive portal is used in conjunction with other technologies that make ups the network infrastructure. A brief description of the infrastructure for a typical network with a captive portal is shown in figure **??**.

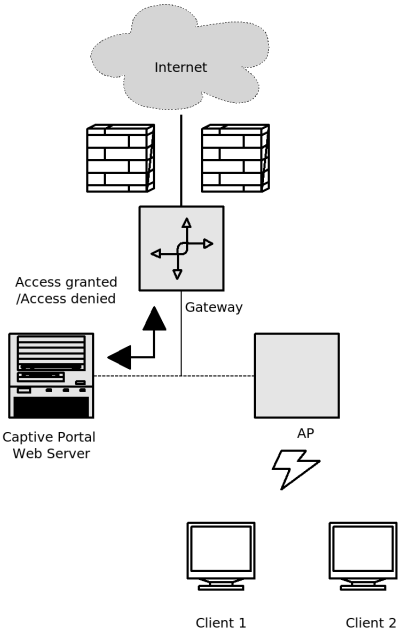

Figure 1. Typical infrastructure of a network with a captive portal

In this network structure one or many clients can connect to a specific access point, AP, that redirects their traffic to a gateway that controls access to the network. If a client is not granted access the gateway will redirect its traffic to a captive portal webserver where a local web page is stored which turns the client web browser into a tool for authentication, paying or acceptance of terms [**?**]. The outcome of this process will then be sent back to the gateway and if access granted the client will be redirected to the router and potentially to some sort of network, internal or external.

3.3. How a Captive Portal is Implemented

A captive portal solution can be implemented in a number of ways but three common ways of implementation are *redirection by HTTP*, *IP redirection* and *redirection by DNS*. [**?**]

In the first way of implementation, redirection by HTTP, which also is shown in figure **??**. Any HTTP website request by a client is intercepted by a firewall and forwarded to a gateway or redirect server which responds with a regular HTTP response containing HTTP status code 302[1] to redirect the client to a captive portal web server. [?]

1. HTTP status code 302 means that the requested address has been found but it resides under a different URI

The second common way of implementation, IP address redirection, is redirecting IP data from a client to a alternate IP destination address within the internal network, e.g. from the gateway directly to the IP address of the captive portal web page. This approach is has much in common with the first implementation except that the client browser is not aware that it is redirected to another destination in the internal or external network. [?] [**?**]

Redirection by DNS shown in figure **??**, which is the third way of implementing a captive portal. This uses a strategy where unauthenticated host DNS requests are redirected to a alternate DNS server pointing all of its requests to the IP address of the captive portal web page. [?]

3.4. Security Issues of a Captive Portal

There are some security issues with a captive portal and this is also the reason for why captive portals are considered to be a insufficient security measure and easy to circumvent. The first security issue most captive portals have is that after the a user has been granted access through the gateway a correct IP address and MAC address is really all that it takes in order to use the network. Further more, the IP and MAC address of a user is easy to discover via packet sniffing software such as *Wireshark* [**?**] or *Kismet* [**?**] and when a attacker knows the IP and MAC address of another

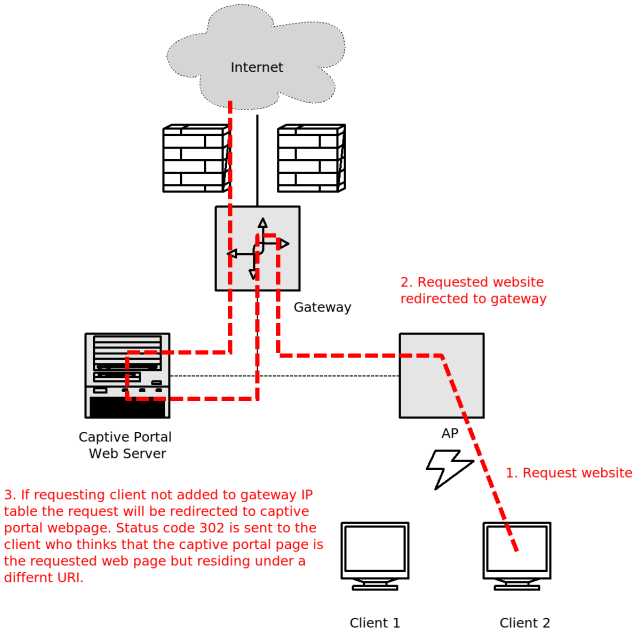

Figure 2. Redirection by HTTP

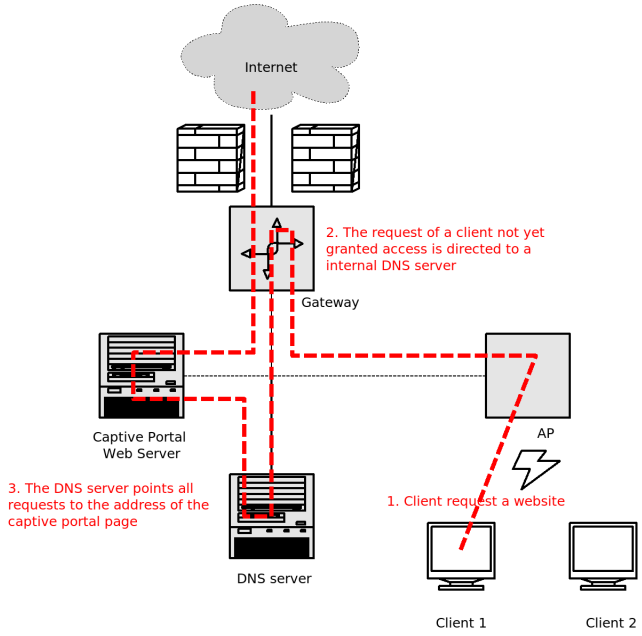

Figure 3. Redirection by DNS

user he or she can easily change his own addresses to match those of the targeted user. [?]

Another issue not directly related to captive portals is the fact that traffic in a public network easily can be eavesdropped if not protected with proper encryption. Encryption can be implemented, but in order for it to be secure and feasible in a network with a large number of users some sort of certificate or similar technology is needed for the key exchange. However, this requires that the user already has a certificate or is connected to a network so that a certificate can be downloaded – a strategy that not might be appropriate in a public network where you want the users to be able to connect easily.

Another option to use instead of certificates would be to use some kind of trusted computing module, TCM, in the connecting computers but as it is right now not many computers have this module and there is also a integrity issue since a user can be linked with his or her TCM.

4. How to Conduct an Attack Against a Captive Portal

In this section we describe how to conduct an attack; what kind of equipment that is needed and if there are certain drivers that needs to be used on for example the network card and we will also give an stepwise description of the attack.

4.1. Hardware and Software Needed for The Attack

The hardware needed is a laptop with a wireless network card. For the disassociate attack there is more requirements. The network card needs to have good support for packet injection. Our attack will use *aircrack-ng* [**?**] and the documentation on supported cards is great. How to install and configure the hardware is a bit of jungle but the forum and wiki on the aircrack-ng homepage has a lot of useful information. In our description we will use Linux but we can't see why it will not work in a other operating system. Other tools that are good to have is a sniffer (Wireshark and Kismet), disassociation software (aircrack-ng suit), and some basic operating system programs like ifconfig and iwconfig.

4.2. Stepwise Description of The Attack

In this example the wireless card is named wlan0

1) Find a wireless access point with a client/victim connected. Take notes on the MAC address for both of them, find out the IP address om the victim and note the channel the access point is running on. In our case we used kismet for this operation.
2) Disconnect your wireless connection and set your MAC address to that of the client.
   ```
   # ifconfig wlan0 down
   # ifconfig wlan0 hw ether <victim
   MAC address>
   ```
3) Put the interface up and running.
   ```
   # ifconfig wlan0 up
   ```
4) Try asking for a IP from the DHCP because it will also set gateway and nameserver else do it by hand.
   ```
   # dhclient wlan0
   ```
5) Test if your internet connection works.
   ```
   $ ping www.google.com
   ```
6) Optional, disassociation of the victim. Start a virtual network card in a monitoring mode.
   ```
   # airmon-ng start wlan0 <channel>
   ```
7) Disassociate the victim, 50 packages was sufficient for us
   ```
   # aireplay -0 50 -a <access point
   address> -c <victims MAC address> -e
   < essid (name of the network)> mon0
   ```

5. Discussion

In this section we discuss how secure a captive portal is and when to use it. We also discuss other possible attacks ideas we have.

We think that captive portal is a low security feature that is useful to force a user to authenticate without special programs or certificates. There is almost no setup time for the user and it can have a very user friendly interface with interactive help possibilities, payment and user authentication. However, it will protect a network against free riders but we think that it's not enough protection against more sophisticated attacks. The main reason why we think it is not sufficient protection against more sophisticated attacks is because it operates in the physical layer and the data link layer where protocol spoofing can easily be performed. For example, in public networks most users have root access on their own computers and therefore they can change there IP and MAC address and spoof the addresses of another user. The IP and MAC address of a client computer are also the parameters a captive portal relies on and thereby is this security check easily thwarted.

The fact that most users have root access to their own computers is not only a problem in wireless networks – it is a problem in wired networks as well. However, in a wireless environment it is impossible to detect whether or not a user is entering monitor mode[2] and this makes it hard for a network administrator to detect, trace or prevent a attack against the system.

Captive portals are a low security measure and can be circumvented and except being possible to circumvent a network with this security features has also a problem with non-repudiation of origin [**?**]. It is bad enough for a network provider that users could free ride on their network, but an even worse scenario is if a legitimate users account is used for inappropriate or illegal actions. In this case it can even be hard to prove that the user has been a target of a attack and is innocent Therefore our opinion is that captive portals should be used moderately in sensitive environment such as political or government areas – a user could easily be framed in a whispering campaign.

As we stated in the section introduction a captive portal might be circumvented in other ways than we have described in this paper. One way of circumventing a acaptive portal could be through a man-in-middle attack where a malicious user acts as a rouge access point and forwards the re-labeled requests of a victim. The biggest difference between this strategy and our own is that it probably requires two network cards. A second approach could be to jam the frequencies and connect to the access point with a directional antenna. However, this approach might draw some ill needed

2. When a user is in monitor mode he can eavesdrop on all traffic that is present in the air. By doing this a malicious user can collect data about all the other users and the infrastructure of a network.

attention both by jamming the network for all other users, but also through a suspicious looking antenna.

Despite that captive portals only offers limited protection we think that captive portals in the future still will be in use where peoples want to have really temporary internet connections like on air ports, hotels and coffee shops and not where the user will repeatedly connect.

6. Conclusions

In this final section of this paper we recapitulate the questions that we stated in the beginning. That is, how does a captive portal work, where are its vulnerabilities, what equipment is needed for a attack and can a attack be automated.

a. *How does a captive portal work?* A captive portal is easiest described as a system that holds a unauthenticated user captive until some sort of web based authentication mechanism authenticates the user. Further, the system will intercept all request and force the user to a authentication website and after a successful authentication process the system changes the gateway so that the user is no longer a captive and is redirected to the requested website.

b. *Where are the vulnerabilities in a captive portal and where is the most suitable entry point for a attack?* Our experiments has shown that the most suitable entry point for attacking a captive portal is by spoofing another users IP and MAC address and then connect to the same access point as the victim. However, our experiments did also show that the real benefits when conducting this kind of attacks where low because of network instability related to the fact that there where more than one user with the same IP and MAC address in the network at the same time.

   During our experiments we tried to force the victim to drop his network connection via sending disassociation packages. This was however not a successful strategy since the default setting of most operating systems is to re-associate again if they loose their connection.

   Another problem that we revealed during our experiments was that the attack easily could be thwarted if the victim signed out from the captive portal page - in that case the attackers IP and MAC address where to be removed from the session manager.

c. *What equipment is needed in order to conduct a attack?* As we stated in the beginning of this paper relatively scarce resources are needed in order to conduct a attack against a captive portal. For the "basic" attack there is no special software more then the possibility to change MAC address on the network card. A good network sniffer like kismet can be handy. For the disassociation attack aircrack-ng is needed and there it is some hardware requirements on the network card.

References

[1] Aircrack-ng (2010) *Compatibility Drivers*, <www> [http://www.aircrack-ng.org/doku.php?id=compatibility_drivers], accessed May 12, 2010.

[2] Cisco Systems Inc (2005) *IP Redirect Application Note*, <www> [http://www.cisco.com/en/US/docs/wireless/technology/ip-redirect/technical/reference/ipredir.html], accessed May 12, 2010.

[3] Kershaw, Mike (2010) *Kismet*, <www> [http://www.kismetwireless.net/], accessed May 12, 2010.

[4] Gollman, Dieter (2006) *Computer Security*, John Wiley & Sons Ltd. The Atrium, Southern Gate, Chichester, West Sussex, England, ISBN-13: 978-0-470-86293-3.

[5] SearchMobileComputing.com (2005) *Captive Portals*, <www> [http://searchmobilecomputing.techtarget.com/definition/captive-portal], accessed May 12, 2010.

[6] Virtual Institute of Applied Science (2007) *Captive Portals*, <www> [http://www.vias.org/wirelessnetw/wndw_08_04_04.html], accessed May 12, 2010.

[7] Wikipedia (2010) *Captive Portals*, <www> [http://en.wikipedia.org/wiki/Captive_portal], accessed May 12, 2010.

[8] Wireshark Foundation (2010) *Wireshark*, <www> [http://www.wireshark.org/], accessed May 12, 2010.