简析南京航空航天大學校辦無線局域網「nuaa.portal」的认证方式

Raphael Yang

社会主义核心价值观基金会

開始之前

本文將簡要介紹南京航空航天大學信息化處運營的「nuaa.portal」的身分認證方式與流程。文章結尾將於披露一種模擬「nuaa.portal」的身分認證的實現方法。

「nuaa.portal」建設於 2014 年；在完成僅針對教職員工的測試後，現已對全體學校成員開放。「nuaa」是南航大的英文縮寫，「portal」意為「門戶」。通過這個門戶，所有繳清了¥20 每月的網路費的同學，都可以在今年六月一日實施的《網絡安全法》的約束下，享受自由的、公開的互聯網。

「nuaa.portal」在 2.4GHz 與 5GHz ，以 802.11n、802.11ac 兩種制式提供服務；接入點不設密碼，身分認證通過連結後自動跳轉的網頁實現。

認證主要由兩個程序構成：

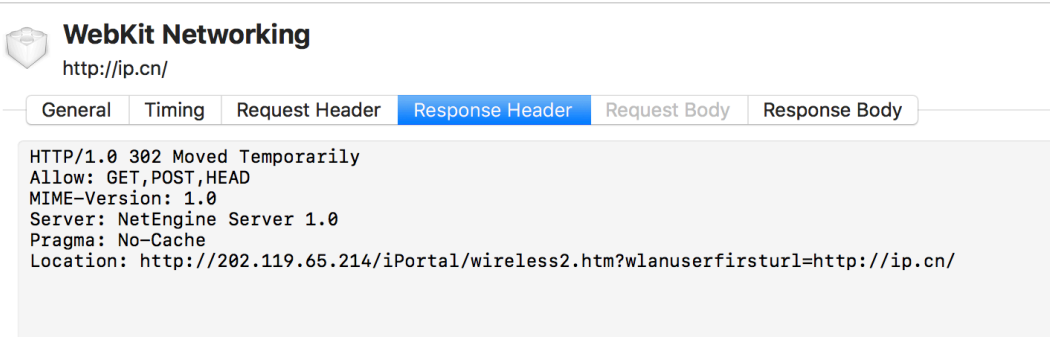
1. 從目標網站跳轉至認證頁面
2. 在認證頁面完成登錄

從目標網站到認證頁面——認證前的跳轉過程

認證從連接上「nuaa.portal」接入點開始。

連接成功後，客戶端將通過 DHCP 程序，得到一個形似「173.31.X.X」的 IP 位址（mask 將為「255.255.224.0」），並獲得「218.104.80.77、202.119.64.123」作為其 DNS 服務器。

此時，由於客戶端尚未通過認證，「nuaa.portal」系統（簡稱認證系統）將通過下文所述的方式將用戶從在瀏覽器中對某一網址的訪問，重定向至認證介面。接下來將以訪問「example.com」為例，描述認證之流程。



1. 用戶在瀏覽器訪問 `http://example.com` 。瀏覽器將通過 GET 方法產生一個請求[1]。受到瀏覽器的請求後，認證系統將會偽造一個代號 302、意味著「Moved Temporarily（臨時異動）」的響應，將讓瀏覽器跳轉至 `http://202.119.65.214/iPortal/wireless2.htm?wlanuserfirsturl=http://example.com/` 。我暫稱其作「重定向頁面甲」。
2. 瀏覽器接下來將再次通過 GET 方法產生一個指向***重定向頁面甲***的請求。此時，認證系統將再次產生一個 302 響應，使瀏覽器跳轉至 `http://202.119.65.214/iPortal/index.htm?from=003cc944be32e25365428f2dd2adbbe2&wlanuserfirsturl=http://example.com/` ，我將它稱作「重定向頁面乙」。

   重定向頁面甲、乙兩者間的差別，僅在於乙較甲多出一個名稱「from」，內容為「003cc944be32e25365428f2dd2adbbe2」的 32-bit hex 的 Query 組件。從它 32 位十六進制數的形式來看，可能是是一段「摘要」。關於它的用途，我有以下猜測：

   2.1 為了唯一地確定上網的設備；

   2.2 為了唯一地確定本次登陸請求。

   令人驚訝的是，經過控制變量的多次試驗，並結合線上檢索結果，我發現「from」的內容恆為 `003cc...`[2]！它或許標記了該系統（南航大校辦無線局域網「nuaa.portal」）的唯一性，以供某些針對此類系統同意開發的應用程式識別運行的環境。

   到此為止，經過兩次重定向，用戶將正式抵達認證頁面；也就是最後一次最後一次抵達的***重定向頁面乙***。

認證介面——在認證頁面實施認證

[1] 1. HTTP 權威指南（中文版），P 51

[2] 幹！它從來沒變過。

「nuaa.portal」的認證頁面地址是：

```
http://202.119.65.214/iPortal/index.htm?from=003cc944be32e25365428f2dd2adbbe2&wlanuserfirsturl=http://www.example.com
```

認證頁面以靜態的方式展示了 IP 位址。

認證頁面主要有兩個行為：

1. 根據 Cookie，向認證系統請求（已保存的）帳號與密碼；

1. 將帳號密碼發送給認證系統，以完成認證。

其中，第一個行為（行為甲）將在網頁載入完成後立即執行；第二個行為（行為乙）將在用戶進行「登陸」操作時觸發。下文將分別介紹這兩個行為。

行為甲——通過 Cookie 從認證系統獲得已保存的帳密

網頁載入完成後，瀏覽器將立即用 POST 方法向認證系統的頁面 `http://202.119.65.214/iPortal/action/doFindUser.do` 發送查詢密碼的請求，瀏覽器將自動附上一組 Cookie（之前登陸時留下的，包括 portal_pass, portal_user & por-

tal_saved 參數）。作為響應，認證系統將會把帳號、密碼、狀態等其它作用未知的信息以 JSON 編寫回傳瀏覽器；瀏覽器最終完成填寫「已保存的密碼」的操作。

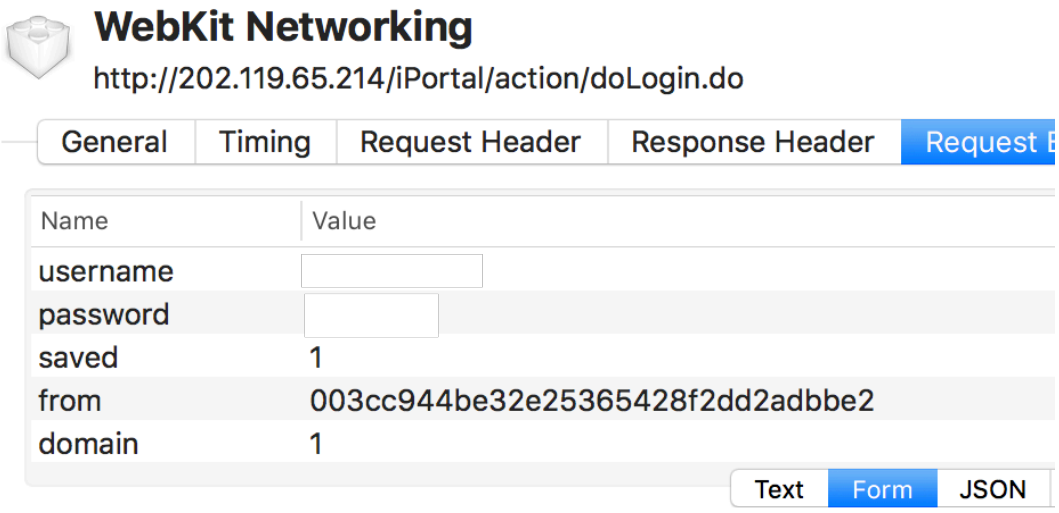
即便沒有此前的 Cookie 留在本地，行為甲仍然會在不附加 Cookie 的情況下進行；這時，認證系統也不會響應以密碼。

另外，若將偽造的一系列 Cookie 附加於該 POST 方法中發送給認證服務，認證服務將會把 Cookie 的原文按照「portal_pass - password」、「portal_user - username」的方式「反射」回來。

行為乙——發送帳號密碼，完成認證

顧名思義，行為乙就是用戶按下「LOGIN」按鈕後發生的行為。與行為甲相同，它也是一個 POST 方法。

用戶名與密碼將先被填入框中。按下「公網資源」按鈕後，瀏覽器將通過 POST 方法產生一個指向 `http://202.119.65.214/iPortal/action/doLogin.do` 的請求。



它的負載是一個表格，包含了 username, password, saved, from & domain 五組資料。其中，username 與 password 即帳號密碼的文本；saved 表明了是否選中了「保存登录信息」復選框，選中時為「1」，否則為「0」；這裡的 from 與介紹***重定向頁面乙***時提到的 from 一致；domain 標示了這個行為是「（登陸）公網資源（域）」、「（登陸）校內資源（域）」與「斷開網絡」中的哪一個。參考 `Application.js`，圖中變量 tt 就是 domain，圖片表明了 tt 的數值與認證狀態的關係。

```
    return;
}
var btns = $('li.btns').html();
if(tt==0){
    $('li.btns').html('<button class="loading" disabled>内网登录中，请稍等..</button>');
}
if(tt==1){
    $('li.btns').html('<button class="loading" disabled>公网登录中，请稍等..</button>');
}
if(tt==-1){
    $('li.btns').html('<button class="loading" disabled>正在断开网络，请稍等..</button>');
}
authing = 1;

$.ajax({
```

針對這個請求，認證服務將會把認證結果以 JSON 編寫回傳網頁；其中的關鍵在於“data”中的“status”。關於可能返回的狀態與其含義，可以參考 `Application.js` 中的提示。


```
var data2 = $.parseJSON(s.data2);
if(data.status=="success"){
    Auth.success(data, data2, tt);
    return false;
} else if(data.status=="reject"){
    $('#password').parent().find('div.error').html('<ul class="error_list"><li>账户或密码错误！！</li></ul>');
    $('#password').addClass('validation-failed');
} else if(data.status=="connected"){
    $('#username').parent().find('div.error').html('<ul class="error_list"><li>您的账户已经登录系统！</li></ul>');
    $('#username').addClass('validation-failed');
} else if(data.status=="authing"){
    $('#username').parent().find('div.error').html('<ul class="error_list"><li>您的账户正在不同的计算机上同时登录！</li></
    $('#username').addClass('validation-failed');
} else if(data.status=="timeout"){
    $('#password').parent().find('div.error').html('<ul class="error_list"><li>登录超时，请稍后再试！</li></ul>');
    $('#password').addClass('validation-failed');
} else if(data.status=="same"){
    $('#password').parent().find('div.error').html('<ul class="error_list"><li>账户和密码不能相同！</li></ul>');
    $('#password').addClass('validation-failed');
} else {
    $('#password').parent().find('div.error').html('<ul class="error_list"><li>登录失败，请稍后再试！</li></ul>');
    $('#password').addClass('validation-failed');
}
```

行為乙中，僅有關於「from」的目的疑問。我針對它在不同條件下進行了測試。

帳號密碼、"from"與認證系統返回狀態的關係		
提供正確的帳密嗎？	提供"003cc94.."作為from嗎？	返回的狀態
是	是	"connected", "success"
是	否，轉而提供"ie"的MD5摘要	片刻後，返回“timeout”
否，轉而提供"null","null"	否，轉而提供"ie"的MD5摘要	"reject"
否，轉而提供"null","null"	是	"reject"
是	提供空字符串	片刻後，返回“timeout”
否，轉而提供"null","null"	提供空字符串	"reject"
是	刪去請求中的from 字段	返回一個標記著“网页不存在。”的網頁
否，轉而提供"null","null"	刪去請求中的from 字段	返回一個標記著“网页不存在。”的網頁

根據實驗結果，我認為認證系統通過『先驗證密碼，後檢查「from」字段內容』的方式實現；當「from」不存在的時候，該請求將不會進入認證系統。

實驗表明，「from」與 `003cc944be32e25365428f2dd2adbbe2` 這個 32-bit hex 對於認證過程不可或缺；但該字段的真正涵義仍撲朔迷離。

至此，「nuaa.portal」的身分認證已經結束。

模擬認證過程

在探索期間，我學習使用了許多新的方法，包括 UNIX 的 cURL、Python 的 Requests 模組等。此外，我還實現了一台能夠做到永遠在線的路由器。遲些時候，我會單獨作文介紹實現它的過程。

- 使用 cURL 實現認證過程

Shell Script 代碼如下。

```
curl 'http://202.119.65.214/iPortal/action/doLogin.do' \

-XPOST \

-H 'Content-Type: application/x-www-form-urlencoded' \

-H 'Referer:
http://202.119.65.214/iPortal/index.htm?from=003cc944be32e25365428f2dd2adbbe2&
wlanuserfirsturl=http://www.ip.cn/' \

-H 'Accept: application/json, text/javascript, */*; q=0.01' \

-H 'User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_12_6) AppleWeb-
Kit/603.3.8 (KHTML, like Gecko) Version/10.1.2 Safari/603.3.8' \

-H 'Origin: http://202.119.65.214' \

-H 'X-Requested-With: XMLHttpRequest' \

--data 'username=[你的用戶名]&password=[你的密碼]&saved=0&from=&domain=1'
```

透過 cURL，你可以很方便地在終端內登陸「nuaa.portal」。

- 透過 Python 實現

我參考了@lmscsomg 的 HackCompusWifi 項目代碼。具體的文件在這裡：HackCompusWifi/test.py，你可以通過修改其中的“username”和“password”後使用。

- 實現於 iOS 設備上（毋須打開 Safari）的一鍵登陸

參考@lmscsomg 的代碼，我在 Pythonista3 中實現了一個 Widget 小程序，達成了在 iOS 通知中心小工具中直接登陸「nuaa.portal」的功能；這段代碼，你可以在我的 GitHub 主頁倉庫中找到[3]。

此外，我還編寫了一個 Workflow 工作流小程序，你可以從這裡得到這個它；你可能是第一次接觸 Workflow 這個 iOS 上的應用程序。我認為它的用戶介面友好、功能強大，並且擁有許多唾手可得的資源，因此在這裡一併向你推薦之。

你可以從這裡下載 Workflow 主程序。我還建議你在網上檢索 Workflow 愛好者為其編寫的說明文檔和有趣的工作流小程序；你在百度搜索的框框里鍵入 `site:sspai.com Workflow` 後換行應該就會有不少收穫。

總結

本文簡要介紹了南京航空航天大學信息化處運營的無線局域網「nuaa.portal」的身分認證方式與過程。從訪問鍵入的目標網站開始，經過共兩次跳轉後抵達認證頁面。在認證頁面，網頁將在完成載入後立即用 Cookies 向認證系統請求密碼以完成「保存密碼」功能。當用戶輸入帳號密碼後點擊登陸按鈕，網頁將附加帳號密碼於請求中發向認證系統；認證系統將根據請求內容回應認證狀態。「nuaa.portal」的身分認證至此結束。

從上文針對驗證過程的分析，我們很容易就可以找到許多「nuaa.portal」的不安全因素和匪夷所思的設定。

關於安全：最基本的——「nuaa.portal」是個開放的無線局域網，這意味著所有訊息都將被在空中明文發送！或許我們要擔心一下「HackRF」（或是別的更簡單的嗅探工具！）。在認證過程中，用戶的密碼將來回被發送許多次。

至於匪夷所思的設定：首先是上文不斷提到的，又是重定向期間又是 POST 方法裡的「from」字段，還有那段 32 位十六進制數——我仍然想不明白它發揮作用，除了浪費網絡帶寬；或許是為將來的某個功能預留一個接口？如果你有什麼想法，請

[3] 我是個 GitHub 新手，正在學習 Git；除了這個代碼之外，我還有許多小項目。我將一一把它們陳列在我的主頁下！

將你的思路和我一起討論。:) 其次，我覺得認證前的重定向過程，加入「重定向頁面甲」的設定完全是多此一舉——原因還是歸結到「from」永遠是個常量；如此的跳轉完全可以在一次中實現，進而節約下層連結中握手的時間。

除此之外，「nuaa.portal」還有包括「不能為蘋果設備所辨認為需要登錄的熱點」等小問題，在此不再贅述。

雖然有許多毛病，但「nuaa.portal」仍然能發揮它的作用。在南航大的宿舍，此前沒有一種互聯網接入手段，是能夠滿足「包月計費、無需按時更換密碼、不限制流量」的條件的。通過文中的分析，我設計實現了一小小的、通過 nuaa.portal 接入的互聯網的路由器；或者說，文中的探索，只是這個實驗中小小的一個步驟。遲些時候，我講作文講述我的靈感和實現過程。

附錄

- 本文使用到的工具軟件

1. Surge
2. cURL

- 探索時候發現的新方法

1. Safari 調試
2. Surge 抓包

- 本文所用參考資料的**檢索關鍵詞**

1. HTML 語法
2. w3schools
3. 菜鳥教程，Python

寫在最後

感謝您閱讀我的文章！文章乃一人之力所作，若有表述不清或錯字、別字之處，歡迎您指出與我聯繫。若您有新的點子或靈感，我將更加開心能夠與您溝通交流，產生新的想法！

再次感謝您的來訪！

P.S.. 這是我的聯繫方式：

Raphael.Yang@gms.tku.edu.tw

向您推薦電子郵件作為我們的聯繫方式是有原因的……