XSS exists in the seacms V6.61 site name and can be executed by setting up the paylod execution script.(exp:https://github.com/MichaelWayneLIU/seacms/blob/master/seacms.md)

seacms V6.61 has XSS via the site name parameter on an adm1n/admin\_config.php page (aka a system management page).

There is two CSRF vulnerability that can add the administrator account

After the administrator logged in, open the following two page  
poc：  
one.html---add a user

two.html---add a admin

After the administrator logged in, open the following two csrf page.The administrator will add a user unknown.(exp:https://github.com/MichaelWayneLIU/seacms/blob/master/seacms1.md)

An issue was discovered in seacms 6.61. There is a CSRF vulnerability that can add a user account via adm1n/admin\_manager.php?action=add.

After the administrator logged in, open the following two csrf page.The administrator will add a user to admin group unknown.(exp:https://github.com/MichaelWayneLIU/seacms/blob/master/seacms1.md)

An issue was discovered in seacms 6.61. There is a CSRF vulnerability that can add an admin account via adm1n/admin\_manager.php?action=save&id=2.

Remote Code execution

.

After the admin logged in,visit http://127.0.0.1/seacms/adm1n/admin\_ip.php to set the allowed ip as special value,we will get a webshell.(exp:https://github.com/MichaelWayneLIU/seacms/blob/master/seacms2.md)

SeaCMS v6.61 allows Remote Code execution by placing PHP code in a allowed ip address (aka ip) to /admin/admin\_ip.php (aka /adm1n/admin\_ip.php). The code is executed by visiting adm1n/admin\_ip.php or data/admin/ip.php. This can also be exploited through CSRF.

<https://github.com/MichaelWayneLIU/seacms/blob/master/seacms2.md>

<http://127.0.0.1/seacms/adm1n/admin_template.php?action=del&filedir=../templets/default/html/../../../install/install_lock.txt>

<http://127.0.0.1/seacms/adm1n/admin_database.php?action=redat&delfile=1&redStruct=1&bakfiles=/../../../install/install_lock.txt>

<http://127.0.0.1/seacms/adm1n/admin_topic_vod.php?tid=-1%20union%20select%20(extractvalue(1,concat(0x7e,(select%20user()),0x7e))),2,3,4,5,6,7,8,9>

<http://127.0.0.1/seacms/adm1n/admin_reslib.php?action=day&rid=ataoju.com&pg=31&url=http://207.148.74.197:8888&backurl=admin_reslib.php>

After the admin logged in,visit [http://127.0.0.1/seacms/adm1n/admin\_reslib.php?action=day&rid=ataoju.com&pg=31&url=http://{server}:8888&backurl=admin\_reslib.php](http://127.0.0.1/seacms/adm1n/admin_reslib.php?action=day&rid=ataoju.com&pg=31&url=http://%7bserver%7d:8888&backurl=admin_reslib.php), The system will initiate a url request to address{server}.(exp:https://github.com/MichaelWayneLIU/seacms/blob/master/seacms3.md)

Seacms's adm1n/admin\_reslib.php has an SSRF vulnerability. The attacker is allowed to initiate a request to the intranet host, obtain the banner information of the web application reachable server service, and collect the fingerprint identification of the intranet web application.

After the admin logged in,visit <http://127.0.0.1/seacms/adm1n/admin_topic_vod.php?tid=-1%20union%20select%20(extractvalue(1,concat(0x7e,(select%20user()),0x7e))),2,3,4,5,6,7,8,9>

, the current user of the database is obtained through SQL injection..(exp:https://github.com/MichaelWayneLIU/seacms/blob/master/seacms4.md)

An issue was discovered in seacms through V6.61. SQL injection exists via the tid parameter in an adm1n/admin\_topic\_vod.php ?tid=1 request.

Arbitrary file deletion

After the admin logged in,visit http://127.0.0.1/seacms/adm1n/admin\_database.php?action=redat&delfile=1&redStruct=1&bakfiles=/../../../install/install\_lock.txt, the lock file install\_lock.txt is deleted.(exp:https://github.com/MichaelWayneLIU/seacms/blob/master/seacms5.md)

An issue was discovered in seacms through V6.61. adm1n/admin\_database.php allows remote attackers to delete arbitrary files via directory traversal sequences in the bakfiles parameter. This will cause the system to reinstall by deleting install\_lock.txt.