Product Solutions Resources Open Source Enterprise Pricing Sign in Sign up

Tib3rius / AutoRecon Public

Notifications Fork 877 Star 5.2k

Code Issues 44 Pull requests 3 Actions Projects Wiki Security Insights

main Go to file Code

282 Commits

.github
autorecon
.gitignore
Dockerfile
LICENSE
README.md
autorecon.py
pyproject.toml
requirements.txt

README GPL-3.0 license

> It's like bowling with bumpers. - @ippsec

AutoRecon

AutoRecon is a multi-threaded network reconnaissance tool which performs automated enumeration of services. It is intended as a time-saving tool for use in CTFs and other penetration testing environments (e.g. OSCP). It may also be useful in real-world engagements.

The tool works by firstly performing port scans / service detection scans. From those initial results, the tool will launch further enumeration scans of those services using a number of different tools. For example, if HTTP is found, feroxbuster will be launched (as well as many others).

Everything in the tool is highly configurable. The default configuration performs **no automated exploitation** to keep the tool in line with OSCP exam rules. If you wish to add automatic exploit tools to the configuration, you do so at your own risk. The author will not be held responsible for negative actions that result from the mis-use of this tool.

Disclaimer: While AutoRecon endeavors to perform as much identification and enumeration of services as possible, there is no guarantee that every service will be identified, or that every service will be fully enumerated. Users of AutoRecon (especially students) should perform their own manual enumeration alongside AutoRecon. Do not rely on this tool alone for exams, CTFs, or other engagements.

Origin

About

AutoRecon is a multi-threaded network reconnaissance tool which performs automated enumeration of services.

Readme
GPL-3.0 license
Activity
5.2k stars
102 watching
877 forks
Report repository

Releases

No releases published

Sponsor this project

https://www.buymeacoffee.com/tib3rius

Learn more about GitHub Sponsors

Packages

No packages published

Contributors 25

+ 11 contributors

Languages

Python 99.6% Dockerfile 0.4%

AutoRecon was inspired by three tools which the author used during the OSCP labs: Reconnoitre, ReconScan, and bscan. While all three tools were useful, none of the three alone had the functionality desired. AutoRecon combines the best features of the aforementioned tools while also implementing many new features to help testers with enumeration of multiple targets.

Features

- Supports multiple targets in the form of IP addresses, IP ranges (CIDR notation), and resolvable hostnames. IPv6 is also supported.
- Can scan multiple targets concurrently, utilizing multiple processors if they are available.
- Advanced plugin system allowing for easy creation of new scans.
- Customizable port scanning plugins for flexibility in your initial scans.
- Customizable service scanning plugins for further enumeration.
- Suggested manual follow-up commands for when automation makes little sense.
- Ability to limit port scanning to a combination of TCP/UDP ports.
- Ability to skip port scanning phase by supplying information about services which should be open.
- Global and per-scan pattern matching which highlights and extracts important information from the noise.
- An intuitive directory structure for results gathering.
- Full logging of commands that were run, along with errors if they fail.
- A powerful config file lets you use your favorite settings every time.
- A tagging system that lets you include or exclude certain plugins.
- Global and per-target timeouts in case you only have limited time.
- Four levels of verbosity, controllable by command-line options, and during scans using Up/Down arrows.
- Colorized output for distinguishing separate pieces of information. Can be turned off for accessibility reasons.

Installation

There are three ways to install AutoRecon: pipx, pip, and manually. Before installation using any of these methods, certain requirements need to be fulfilled. If you have not refreshed your apt cache recently, run the following command so you are installing the latest available packages:

```
sudo apt update
```

Python 3

AutoRecon requires the usage of Python 3.8+ and pip, which can be installed on Kali Linux using the following commands:

```
sudo apt install python3
sudo apt install python3-pip
```

Supporting Packages

Several commands used in AutoRecon reference the SecLists project, in the directory /usr/share/seclists/. You can either manually download the SecLists project to this directory (https://github.com/danielmiessler/SecLists), or if you are using Kali Linux (**highly recommended**) you can run the following commands:

```
sudo apt install seclists
```

AutoRecon will still run if you do not install SecLists, though several commands may fail, and some manual commands may not run either.

Additionally the following commands may need to be installed, depending on your OS:

```
curl
dnsrecon
enum4linux
feroxbuster
gobuster
impacket-scripts
nbtscan
nikto
nmap
onesixtyone
oscanner
redis-tools
smbclient
smbmap
snmpwalk
sslscan
svwar
tnscmd10g
whatweb
wkhtmltopdf
```

On Kali Linux, you can ensure these are all installed using the following commands:

```
sudo apt install seclists curl dnsrecon enum4linux feroxbuster gobus
```

Installation Method #1: pipx (Recommended)

It is recommended you use `pipx` to install AutoRecon. pipx will install AutoRecon in it's own virtual environment, and make it available in the global context, avoiding conflicting package dependencies and the resulting instability. First, install pipx using the following commands:

```
sudo apt install python3-venv
python3 -m pip install --user pipx
python3 -m pipx ensurepath
```

You will have to re-source your ~/.bashrc or ~/.zshrc file (or open a new tab) after running these commands in order to use pipx.

Install AutoRecon using the following command:

```
pipx install git+https://github.com/Tib3rius/AutoRecon.git
```

Note that if you want to run AutoRecon using sudo (required for faster SYN scanning and UDP scanning), you have to use *one* of the following examples:

```
sudo env "PATH=$PATH" autorecon [OPTIONS]
sudo $(which autorecon) [OPTIONS]
```

Installation Method #2: pip

Alternatively you can use `pip` to install AutoRecon using the following command:

```
python3 -m pip install git+https://github.com/Tib3rius/AutoRecon.git
```

Note that if you want to run AutoRecon using sudo (required for faster SYN scanning and UDP scanning), you will have to run the above command as the root user (or using sudo).

Similarly to `pipx`, if installed using `pip` you can run AutoRecon by simply executing `autorecon`.

Installation Method #3: Manually

If you'd prefer not to use `pip` or `pipx`, you can always still install and execute `autorecon.py` manually as a script. From within the AutoRecon directory, install the dependencies:

```
python3 -m pip install -r requirements.txt
```

You will then be able to run the `autorecon.py` script:

```
python3 autorecon.py [OPTIONS] 127.0.0.1
```

Upgrading

pipx

Upgrading AutoRecon when it has been installed with pipx is the easiest, and is why the method is recommended. Simply run the following command:

```
pipx upgrade autorecon
```

pip

If you've installed AutoRecon using pip, you will first have to uninstall AutoRecon and then re-install using the same install command:

```
python3 -m pip uninstall autorecon
python3 -m pip install git+https://github.com/Tib3rius/AutoRecon.git
```

Manually

If you've installed AutoRecon manually, simply change to the AutoRecon directory and run the following command:

```
git pull
```

Assuming you did not modify any of the content in the AutoRecon directory, this should pull the latest code from this GitHub repo, after which you can run AutoRecon using the autorecon.py script as per usual.

Plugins

A plugin update process is in the works. Until then, after upgrading, remove the ~/.local/share/AutoRecon directory and run AutoRecon with any argument to repopulate with the latest files.

Usage

AutoRecon uses Python 3 specific functionality and does not support Python 2.

```
usage: autorecon [-t TARGET_FILE] [-p PORTS] [-m MAX_SCANS] [-mp MAX_
                 [--exclude-tags TAGS] [--port-scans PLUGINS] [--ser
                 [--add-plugins-dir PLUGINS_DIR] [-l [TYPE]] [-o OUT
                 [--heartbeat HEARTBEAT] [--timeout TIMEOUT] [--targ
                 [--proxychains] [--disable-sanity-checks] [--disabl
                 [-v] [--version] [--curl.path VALUE] [--dirbuster.t
                 [--dirbuster.wordlist VALUE [VALUE ...]] [--dirbust
```

```
                  [--onesixtyone.community-strings VALUE] [--global.u
                  [--global.domain VALUE] [-h]
                  [targets ...]

Network reconnaissance tool to port scan and automatically enumerate

positional arguments:
  targets               IP addresses (e.g. 10.0.0.1), CIDR notation

optional arguments:
  -t TARGET_FILE, --target-file TARGET_FILE
                        Read targets from file.
  -p PORTS, --ports PORTS
                        Comma separated list of ports / port ranges
                        TCP/UDP, put port(s) at start or specify B:
  -m MAX_SCANS, --max-scans MAX_SCANS
                        The maximum number of concurrent scans to ru
  -mp MAX_PORT_SCANS, --max-port-scans MAX_PORT_SCANS
                        The maximum number of concurrent port scans
  -c CONFIG_FILE, --config CONFIG_FILE
                        Location of AutoRecon's config file. Default
  -g GLOBAL_FILE, --global-file GLOBAL_FILE
                        Location of AutoRecon's global file. Default
  --tags TAGS           Tags to determine which plugins should be in
                        groups with a comma (,) to create multiple g
                        at least one group. Default: default
  --exclude-tags TAGS   Tags to determine which plugins should be ex
                        groups with a comma (,) to create multiple g
                        at least one group. Default: None
  --port-scans PLUGINS  Override --tags / --exclude-tags for the lis
  --service-scans PLUGINS
                        Override --tags / --exclude-tags for the lis
  --reports PLUGINS     Override --tags / --exclude-tags for the lis
  --plugins-dir PLUGINS_DIR
                        The location of the plugins directory. Defau
  --add-plugins-dir PLUGINS_DIR
                        The location of an additional plugins direct
  -l [TYPE], --list [TYPE]
                        List all plugins or plugins of a specific ty
  -o OUTPUT, --output OUTPUT
                        The output directory for results. Default: r
  --single-target       Only scan a single target. A directory named
                        be created within the output directory. Defa
  --only-scans-dir      Only create the "scans" directory for result
                        Default: False
  --no-port-dirs        Don't create directories for ports (e.g. sca
                        itself. Default: False
  --heartbeat HEARTBEAT
                        Specifies the heartbeat interval (in seconds
  --timeout TIMEOUT     Specifies the maximum amount of time in minu
  --target-timeout TARGET_TIMEOUT
                        Specifies the maximum amount of time in minu
                        Default: None
  --nmap NMAP           Override the {nmap_extra} variable in scans.
  --nmap-append NMAP_APPEND
                        Append to the default {nmap_extra} variable
  --proxychains         Use if you are running AutoRecon via proxych
  --disable-sanity-checks
                        Disable sanity checks that would otherwise p
  --disable-keyboard-control
                        Disables keyboard control ([s]tatus, Up, Dow
  --force-services SERVICE [SERVICE ...]
                        A space separated list of services in the fo
  --accessible          Attempts to make AutoRecon output more acces
  -v, --verbose         Enable verbose output. Repeat for more verbo
  --version             Prints the AutoRecon version and exits.
  -h, --help            Show this help message and exit.

plugin arguments:
  These are optional arguments for certain plugins.

  --curl.path VALUE     The path on the web server to curl. Default:
  --dirbuster.tool {feroxbuster,gobuster,dirsearch,ffuf,dirb}
                        The tool to use for directory busting. Defau
  --dirbuster.wordlist VALUE [VALUE ...]
```

```
                        The wordlist(s) to use when directory bustin
                        ['~/.local/share/AutoRecon/wordlists/dirbust
  --dirbuster.threads VALUE
                        The number of threads to use when directory
  --dirbuster.ext VALUE
                        The extensions you wish to fuzz (no dot, com
  --onesixtyone.community-strings VALUE
                        The file containing a list of community stri
                        community-strings-onesixtyone.txt

global plugin arguments:
  These are optional arguments that can be used by all plugins.

  --global.username-wordlist VALUE
                        A wordlist of usernames, useful for brutefor
  --global.password-wordlist VALUE
                        A wordlist of passwords, useful for brutefor
  --global.domain VALUE
                        The domain to use (if known). Used for DNS a
```

Verbosity

AutoRecon supports four levels of verbosity:

- (none) Minimal output. AutoRecon will announce when scanning targets starts / ends.
- (-v) Verbose output. AutoRecon will additionally announce when plugins start running, and report open ports and identified services.
- (-vv) Very verbose output. AutoRecon will additionally specify the exact commands which are being run by plugins, highlight any patterns which are matched in command output, and announce when plugins end.
- (-vvv) Very, very verbose output. AutoRecon will output everything. Literally every line from all commands which are currently running. When scanning multiple targets concurrently, this can lead to a ridiculous amount of output. It is not advised to use -vvv unless you absolutely need to see live output from commands.

Note: You can change the verbosity of AutoRecon mid-scan by pressing the up and down arrow keys.

Results

By default, results will be stored in the ./results directory. A new sub directory is created for every target. The structure of this sub directory is:

```
.
├── exploit/
├── loot/
├── report/
│   ├── local.txt
│   ├── notes.txt
│   ├── proof.txt
│   └── screenshots/
└── scans/
        ├── _commands.log
        ├── _manual_commands.txt
        ├── tcp80/
        ├── udp53/
        └── xml/
```

The exploit directory is intended to contain any exploit code you download / write for the target.

The loot directory is intended to contain any loot (e.g. hashes, interesting files) you find on the target.

The report directory contains some auto-generated files and directories that are useful for reporting:

- local.txt can be used to store the local.txt flag found on targets.
- notes.txt should contain a basic template where you can write notes for each service discovered.
- proof.txt can be used to store the proof.txt flag found on targets.
- The screenshots directory is intended to contain the screenshots you use to document the exploitation of the target.

The scans directory is where all results from scans performed by AutoRecon will go. This includes port scans / service detection scans, as well as any service enumeration scans. It also contains two other files:

- _commands.log contains a list of every command AutoRecon ran against the target. This is useful if one of the commands fails and you want to run it again with modifications.
- _manual_commands.txt contains any commands that are deemed "too dangerous" to run automatically, either because they are too intrusive, require modification based on human analysis, or just work better when there is a human monitoring them.

By default, directories are created for each open port (e.g. tcp80, udp53) and scan results for the services found on those ports are stored in their respective directories. You can disable this behavior using the --no-port-dirs command line option, and scan results will instead be stored in the scans directory itself.

If a scan results in an error, a file called _errors.log will also appear in the scans directory with some details to alert the user.

If output matches a defined pattern, a file called _patterns.log will also appear in the scans directory with details about the matched output.

The scans/xml directory stores any XML output (e.g. from Nmap scans) separately from the main scan outputs, so that the scans directory itself does not get too cluttered.

Testimonials

> AutoRecon was invaluable during my OSCP exam, in that it saved me from the tedium of executing my active information gathering commands myself. I was able to start on a target with all of the information I needed clearly laid in front of me. I would strongly recommend this utility for anyone in the PWK labs, the OSCP exam, or other environments such as VulnHub or HTB. It is a great tool for both people just starting down their journey into OffSec and seasoned veterans alike. Just make sure that somewhere between those two points you take the time to learn what's going on "under the hood" and how / why it scans what it does.
>
> - b0ats (rooted 5/5 exam hosts)

> Wow, what a great find! Before using AutoRecon, ReconScan was my goto enumeration script for targets because it automatically ran the enumeration commands after it finds open ports. The only thing missing was the automatic creation of key directories a pentester might need during an engagement (exploit, loot, report, scans). Reconnoitre did this but didn't automatically run those commands for you. I thought ReconScan that was the bee's knees until I gave AutoRecon a try. It's awesome! It combines the best features of Reconnoitre (auto directory creation) and ReconScan (automatically executing the enumeration commands). All I have to do is run it on a target or a set of targets and start going over the information it has already collected while it continues the rest of scan. The proof is in the pudding :) Passed the OSCP exam! Kudos to Tib3rius!
>
> - werk0ut

> A friend told me about AutoRecon, so I gave it a try in the PWK labs. AutoRecon launches the common tools we all always use, whether it be nmap or nikto, and also creates a nice subfolder system based on the targets you are attacking. The strongest feature of AutoRecon is the speed; on the OSCP exam I left the tool

> running in the background while I started with another target, and in a matter of minutes I had all of the AutoRecon output waiting for me. AutoRecon creates a file full of commands that you should try manually, some of which may require tweaking (for example, hydra bruteforcing commands). It's good to have that extra checklist.
>
> - tr3mb0 (rooted 4/5 exam hosts)

> Being introduced to AutoRecon was a complete game changer for me while taking the OSCP and establishing my penetration testing methodology. AutoRecon is a multi-threaded reconnaissance tool that combines and automates popular enumeration tools to do most of the hard work for you. You can't get much better than that! After running AutoRecon on my OSCP exam hosts, I was given a treasure chest full of information that helped me to start on each host and pass on my first try. The best part of the tool is that it automatically launches further enumeration scans based on the initial port scans (e.g. run enum4linux if SMB is detected). The only bad part is that I did not use this tool sooner! Thanks Tib3rius.
>
> - rufy (rooted 4/5 exam hosts)

> AutoRecon allows a security researcher to iteratively scan hosts and identify potential attack vectors. Its true power comes in the form of performing scans in the background while the attacker is working on another host. I was able to start my scans and finish a specific host I was working on - and then return to find all relevant scans completed. I was then able to immediately begin trying to gain initial access instead of manually performing the active scanning process. I will continue to use AutoRecon in future penetration tests and CTFs, and highly recommend you do the same.
>
> - waar (rooted 4.99/5 exam hosts)

> "If you have to do a task more than twice a day, you need to automate it." That's a piece of advice that an old boss gave to me. AutoRecon takes that lesson to heart. Whether you're sitting in the exam, or in the PWK labs, you can fire off AutoRecon and let it work its magic. I had it running during my last exam while I worked on the buffer overflow. By the time I finished, all the enum data I needed was there for me to go through. 10/10 would recommend for anyone getting into CTF, and anyone who has been at this a long time.
>
> - whoisflynn

> I love this tool so much I wrote it.
>
> - Tib3rius (rooted 5/5 exam hosts)

> I highly recommend anyone going for their OSCP, doing CTFs or on HTB to checkout this tool. Been using AutoRecon on HTB for a month before using it over on the PWK labs and it helped me pass my OSCP exam. If you're having a hard time getting settled with an enumeration methodology I encourage you to follow the flow and techniques this script uses. It takes out a lot of the tedious work that you're probably used to while at the same time provide well-organized subdirectories to quickly look over so you don't lose your head. The manual commands it provides are great for those specific situations that need it when you have run out of options. It's a very valuable tool, cannot recommend enough.
>
> - d0hnuts (rooted 5/5 exam hosts)

> Autorecon is not just any other tool, it is a recon correlation framweork for engagements. This helped me fire a whole bunch of scans while I was working on other targets. This can help a lot in time management. This assisted me to own 4/5 boxes in pwk exam! Result: Passed!
>
> - Wh0ami (rooted 4/5 exam hosts)

> The first time I heard of AutoRecon I asked whether I actually needed this, my enumeration was OK... I tried it with an open mind and straight away was a little

> floored on the amount of information that it would generate. Once I got used to it, and started reading the output I realized how much I was missing. I used it for the OSCP exam, and it found things I would never have otherwise found. I firmly believe, without AutoRecon I would have failed. It's a great tool, and I'm very impressed what Tib3rius was able to craft up. Definitely something I'm already recommending to others, including you!
>
> - othornew

> AutoRecon helped me save valuable time in my OSCP exam, allowing me to spend less time scanning systems and more time breaking into them. This software is worth its weight in gold!
>
> - TorHackr

> The magical tool that made enumeration a piece of cake, just fire it up and watch the beauty of multi-threading spitting a ton of information that would have taken

© 2024 GitHub, Inc. Terms Privacy Security Status Docs Contact Manage cookies Do not share my personal information