Active Reconnaissance

SoftUni Team

Technical Trainers

Software University

SoftUni

Software University

https://softuni.bg

Have a Questions?

Software University

sli.do

#Cyber-Security

Table of Contents

1. Active Reconnaissance Theory
2. Active Reconnaissance Techniques
3. Active Reconnaissance Tools

Active Reconnaissance Theory

Active Reconnaissance in a Nutshell

Software University

- **Active Reconnaissance** is the act of gathering detailed information about the target
- This can include things like: port, service, version and more
- **Active Reconnaissance** is also called scanning or enumeration, since each technique is actively generating traffic (in form of packets) and the results lies on the output
- In a nutshell, every scanner sends specifically crafted packets, and based on their responses, it can determine the desired results
- **Active Reconnaissance** usually is the next step after the passive one
  - The idea here is to collect more detailed information about the target
  - Use that information for basic context for vulnerabilities

Active Reconnaissance Techniques

Port Scanning

- **Port scanning** is one of the main enumeration techniques on every engagement
- It is usually done with nmap
- Port scanning is sending packets on different ports, and based on the output determines if the port is open, closed and filtrated
  - While open and close states are self explanatory, filtrated means that the port is behind a firewall, and the scanner cannot determine it's state

Port Scanning

- Port scanners like nmap can send packets on both transport layers (**TCP**, **UDP**)
- Port scanners like nmap have extended functionality, with given flags it can determine the running services on the opened ports
- If nmap fails to enumerate service and version, the port must be manually enumerated with netcat(nc)

Service Enumeration

- The idea of having opened ports is to host services on it
- Mostly encountered services are:
  - **FTP** (Port 21)
  - **HTTP/S** (Port 80, 443)
  - **SSH** (Port 22)
  - **HTTP/S Proxy** (Port 8080, 8443)
  - **SMTP** (Port 25)

Service Enumeration

- Each service is running its own software, having version and possible vulnerabilities
- It is essential to enumerate every single service on the targeted host, including as much details as possible
- This can be done with nc or specific services (like **BurpSuite**)

Enumerating FTP

- FTP is a service for file sharing, when enumerating the idea is to:
  - Enumerate the **FTP Version**
  - Enumerate the **FTP Configuration** (can we log in anonymously?)
- If both, the version is updated and anon login is disabled, keep ftp at your backpack, you may further compromise credentials, valid for the service

Enumerating SSH

- SSH is genuinely a service we skip, since it is designed secured by default
- When **enumerating SSH**, the idea is to:
  - Enumerate service version
  - From service version, you can enumerate the operating system version

Enumerating HTTP/S

- Most usually, port 80 and 443 are being opened, these are **HTTP/S ports**
- Most likely web applications can be found hosted on them
- Web applications are whole new domain with various of specific (for web) vulnerabilities

Enumerating HTTP/S

Software University

- In a nutshell, when **enumerating web application** we are interested in:
  - Backend / Frontend technology or framework
  - Web server software and its version
  - Mapping the web application (seeing how it works, what it is all about)
  - Finding hidden files and directories
  - Finding hidden gems, like phpinfo, swagger and more
  - Quick wins (hardcoded / default credentials)

Enumerating SMTP

- When **enumerating SMTP**, the idea is to:
  - Enumerate service version
  - Interact with the service, enumerate server name
  - Interact with the service, enumerate valid commands

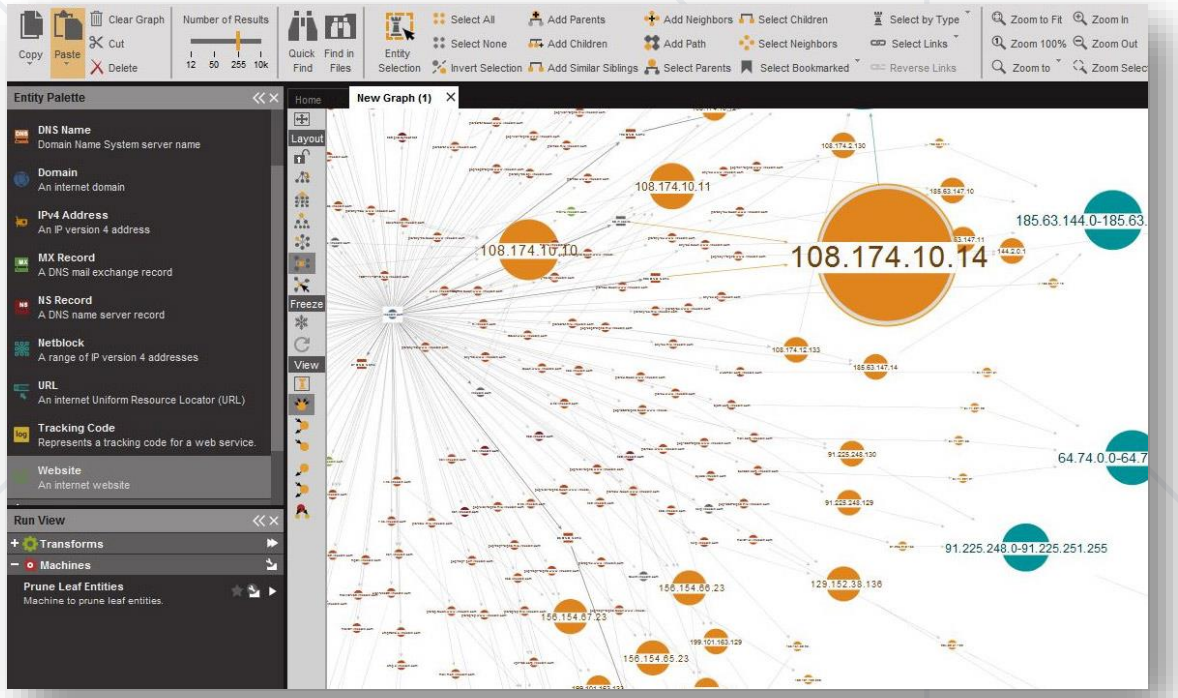
Active Reconnaissance Tools

Discover (All in One)

- Discover (*https://github.com/leebaird/discover*) can be used as active enumeration as well

Maltego

- Maltego (*https://www.maltego.com/*) is software for data collection and visualization. It can perform BOTH passive and active scans, so you must be careful with it
- It can be used as a context or investigation map
- It uses GUI and is easy to use
- It is OS independent



Nmap

- Nmap is the widely used port scanner. Tools from here on out are doing only active enumeraton
- Nmap is wide tool with a lot of options, it can perform **TCP**/**UDP** port enumeration
- It has the ability to:
  - Manually filter what tcp packet to send
  - Perform Firewall Bypassing
  - Scanning for vulnerabilities with its own scripting engine
  - A lot more after the break

Netcat (nc)

- Nc is the **swiss** army knife of the penetration testers
- On its core, **nc** is a software able to connect to any remote service, and send any kind of custom packets
- It can be used for file transfers
- As well as **reverse** / **bind** shells
- And it was invented as a sysadmin tool … (Yes there is a windows version of it too)
- Of course, it was built with **C**

Nikto

- Nikto is a web application scanner
- It scans for:
  - Hidden gems like **.robots.txt**
  - Backup files
  - Basic misconfigurations

```
┌──(kali㉿kali)-[~]
└─$ nikto
- Nikto v2.1.6
---------------------------------------------------------------------------
+ ERROR: No host or URL specified

       -config+            Use this config file
       -Display+           Turn on/off display outputs
       -dbcheck            check database and other key files for syntax erro
rs
       -Format+            save file (-o) format
       -Help               Extended help information
       -host+              target host/URL
       -id+                Host authentication to use, format is id:pass or i
d:pass:realm
       -list-plugins       List all available plugins
       -output+            Write output to this file
       -nossl              Disables using SSL
       -no404              Disables 404 checks
       -Plugins+           List of plugins to run (default: ALL)
       -port+              Port to use (default 80)
       -root+              Prepend root value to all requests, format is /dir
ectory
       -ssl                Force ssl mode on port
       -Tuning+            Scan tuning
       -timeout+           Timeout for requests (default 10 seconds)
       -update             Update databases and plugins from CIRT.net
       -Version            Print plugin and database versions
       -vhost+             Virtual host (for Host header)
                + requires a value

        Note: This is the short help output. Use -H for full help text.
```

Gobuster / dirb

- Gobuster and dirb are tools for directory brute forcing
- Their idea is to scan for hidden directories and files

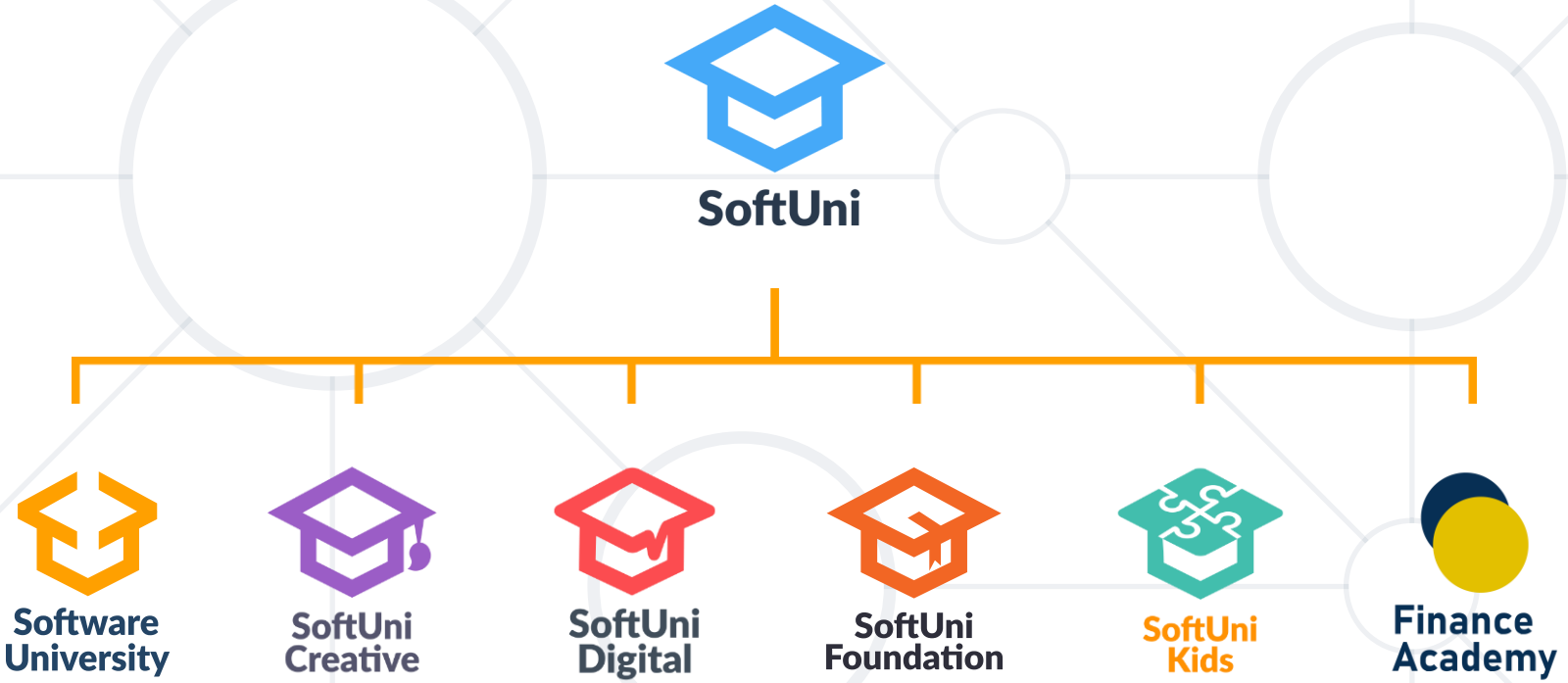
```
Usage:
  gobuster [command]

Available Commands:
  completion  Generate the autocompletion script for the specified shell
  dir         Uses directory/file enumeration mode
  dns         Uses DNS subdomain enumeration mode
  fuzz        Uses fuzzing mode. Replaces the keyword FUZZ in the URL, Headers and the request body
  gcs         Uses gcs bucket enumeration mode
  help        Help about any command
  s3          Uses aws bucket enumeration mode
  version     shows the current version
  vhost       Uses VHOST enumeration mode (you most probably want to use the IP address as the URL parameter)

Flags:
      --delay duration    Time each thread waits between requests (e.g. 1500ms)
  -h, --help              help for gobuster
      --no-color          Disable color output
      --no-error          Don't display errors
  -z, --no-progress       Don't display progress
  -o, --output string     Output file to write results to (defaults to stdout)
  -p, --pattern string    File containing replacement patterns
  -q, --quiet             Don't print the banner and other noise
  -t, --threads int       Number of concurrent threads (default 10)
  -v, --verbose           Verbose output (errors)
  -w, --wordlist string   Path to the wordlist

Use "gobuster [command] --help" for more information about a command.
```

```
_________________
DIRB v2.22
By The Dark Raver
_________________

dirb <url_base> [<wordlist_file(s)>] [options]

========== NOTES ==========
 <url_base> : Base URL to scan. (Use -resume for session resuming)
 <wordlist_file(s)> : List of wordfiles. (wordfile1,wordfile2,wordfile3 ... )

========== HOTKEYS ==========
 'n' -> Go to next directory.
 'q' -> Stop scan. (Saving state for resume)
 'r' -> Remaining scan stats.

========== OPTIONS ==========
 -a <agent_string> : Specify your custom USER_AGENT.
 -b : Use path as is.
 -c <cookie_string> : Set a cookie for the HTTP request.
 -E <certificate> : path to the client certificate.
 -f : Fine tunning of NOT_FOUND (404) detection.
 -H <header_string> : Add a custom header to the HTTP request.
 -i : Use case-insensitive search.
 -l : Print "Location" header when found.
 -N <nf_code>: Ignore responses with this HTTP code.
 -o <output_file> : Save output to disk.
 -p <proxy[:port]> : Use this proxy. (Default port is 1080)
 -P <proxy_username:proxy_password> : Proxy Authentication.
 -r : Don't search recursively.
 -R : Interactive recursion. (Asks for each directory)
 -S : Silent Mode. Don't show tested words. (For dumb terminals)
 -t : Don't force an ending '/' on URLs.
 -u <username:password> : HTTP Authentication.
 -v : Show also NOT_FOUND pages.
 -w : Don't stop on WARNING messages.
 -X <extensions> / -x <exts_file> : Append each word with this extensions.
 -z <millisecs> : Add a milliseconds delay to not cause excessive Flood.

========== EXAMPLES ==========
 dirb http://url/directory/ (Simple Test)
 dirb http://url/ -X .html (Test files with '.html' extension)
 dirb http://url/ /usr/share/dirb/wordlists/vulns/apache.txt (Test with apache.txt wordlist)
 dirb https://secure_url/ (Simple Test with SSL)
```

Summary

- Active Reconnaissance in a **Nutshell**
- Active Reconnaissance **Techniques**
  - **Port Scanning**
  - **Services Enumeration**
- Active Reconnaissance **Tools**

Questions?



SoftUni Diamond Partners

Software University

Trainings @ Software University (SoftUni)

- Software University – High-Quality Education, Profession and Job for Software Developers
  - softuni.bg, about.softuni.bg
- Software University Foundation
  - softuni.foundation
- Software University @ Facebook
  - facebook.com/SoftwareUniversity

License

- This course (slides, examples, demos, exercises, homework, documents, videos and other assets) is **copyrighted content**
- Unauthorized copy, reproduction or use is illegal
- © SoftUni – https://about.softuni.bg/
- © Software University – https://softuni.bg