Lecture 4: Information Gathering

BKACAD's Security Training

Table of Content

Table of Content

Foot-printing Concept

Concept

1st step in the evaluation of the security posture.

Through foot-printing, one can gather maximum information about a computer system or a network and about any devices connected to that network

Table of Content

Passive Information Gathering

Concept

A.K.A Open-source Intelligence/OSINT is the process of collecting openly avaiable information about a target, without any direct interaction with that target.

Passive Information Gathering

Two types

- In the strictest interpretation, NEVER communicate with target directly. Rely on 3rd party for information but wouldn't access any of the target's systems or servers.
  - pros: High level of secrecy
  - cons: Limit results
- In a looser interpretation, might interact with the target, but only as a normal Internet user would.

Passive Information Gathering

Website Recon

If the client has a website, we can gather basic information by simply browsing the site. Small organizations may only have a single website, while large organizations might have many, including some that are not maintained.

This section will be reminded in “Web Security Lecture”.

https://www.megacorpone.com/

Passive Information Gathering

whois Enumeration

Whois is a TCP service, tool, and a type of database that can provide information about a domain name, such as the name server.

https://whois.domaintools.com/

```
kali@kali:~$ whois www.bkacad.com
```

Passive Information Gathering

Exercises

1. Use the `whois` tool in Kali to identify the name servers of MegaCorp One.
2. Code a simple `lookup.py & reverse_lookup.py` script with Python `socket` module, that they can lookup IP from Domain name and reverse.

Hint: `raw_input()`, `gethostbyname()`, `gethostbyaddr()`

Passive Information Gathering

Exercises – Walk through

GNU nano 2.9.3 lookup.py

```
# !/usr/bin/python

'''
Lookup IP from DNS
'''

import socket

def main():
        try:
                target = raw_input("Enter your target DNS: ")
                result = socket.gethostbyname(target)
                print result
        except:
                print "Please check input!"

if __name__ == '__main__':
        main()
```

Passive Information Gathering

Google Hacking

Operator	Syntax	Description
Filetype	filetype:string	Search file with specific type “Ceh” + filetype:pdf
Index of	Index of /string	Display pages with directory listing vulnerability Index of /password
Intitle	intitle:string	Search for pages that contain string in the title intitle:"SQLiteManager" + intext:"Welcome to SQLiteManager version "
Inurl	inurl:string	Display pages within string in the url inurl:/host.txt + filetype:txt + "password"
Info	infor:string	Display information Google stores about the page itself
Link	link:string	Display linked pages based on search term
Site	site:domain	Display pages for specific website or domain holding search term.

Passive Information Gathering

Google Hacking – Demo

Target:

- https://stable.modified-shop.org/.svn/wc.db
- https://stable.modified-shop.org/.svn/pristine/

```
kali@kali:~$ wget http://www.sometarget.tgt/.svn/wc.db
kali@kali:~$ sqlite3 wc.db 'select local_relpath, ".svn/pristine/" ||
substr(checksum,7,2) || "/" || substr(checksum,7) || ".svn-base" as
alpha from NODES;'
```

Passive Information Gathering

Exercises

1. Reproduce pre-demo in Dolcera company with main web-site URL

   https://www.dolcera.com/

2. Make a dork to get all results from Google about this vulnerability

Passive Information Gathering

Netcraft

Netcraft is an Internet services company based in England offering a free web portal that performs various information gathering functions.

https://searchdns.netcraft.com/

https://searchdns.netcraft.com/?restriction=site+contains&host=*.megacorpone.com

Passive Information Gathering

Exercises

1. Use Netcraft to determine what application server is running on www.bkacad.com
2. Code a simple `header_info.py` script with Python `requests` module, that they can check header of target website.

   Hint: `import requests, dir(requests)`

Passive Information Gathering

Exercises – Walk through

GNU nano 2.9.3 header_info.py

```
# !/usr/bin/python

'''
Header checker
'''

import requests

def main():
        try:
                target = raw_input("Enter target website (eg http://abc.com): ")
                response = requests.get(target)
                print response.headers
        except:
                print "Please recheck input!"


if __name__ == "__main__":
        main()
```

Passive Information Gathering

Open-Source Code

One such source of interesting information are open-source projects and online code repositories, such as GitHub, GitLab, and SourceForge.

Code stored online can provide a glimpse into the programming languages and frameworks used by an organization. In some rare occasions, developers have even accidentally committed sensitive data and credentials to public repos.

https://github.com/techgaun/github-dorks

Passive Information Gathering

Shodan

Shodan is a search engine that crawls devices connected to the Internet including but not limited to the World Wide Web. This includes the servers that run websites but also devices like routers and IoT devices.

https://www.shodan.io/

https://help.shodan.io/the-basics/search-query-fundamentals

Passive Information Gathering

Shodan – Demo Attack Un-authenticated ADB

Shodan LIVE demo

```
kali@kali:~$ This is live demo

kali@kali:~$ Nothing inside :’)
```

Table of Content

Active Information Gathering

Concept

Beyond passive information gathering and explore techniques that involve direct interaction with target services.

Main focus on DNS service.

DNS Enumeration

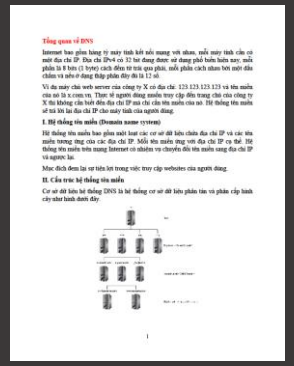
Concept

The Domain Name System (DNS) is one of the most critical systems on the Internet and is a distributed database responsible for translating user-friendly domain names into IP addresses.

This is facilitated by a hierarchical structure that is divided into several zones, starting with the top-level root zone.

www.megacorpone.com

For more information, check DNS-note.pdf file:



DNS Enumeration

DNS process

Process:

1. A hostname of Megacorp-One is entered into a browser or other application.
2. The browser passes the hostname to the operating system's DNS client and the operating system then forwards the request to the external DNS server it is configured to use. This first server in the chain is known as the DNS recursor and is responsible for interacting with the DNS infrastructure and returning the results to the DNS client. The DNS recursor contacts one of the servers in the DNS root zone. The root server then responds with the address of the server responsible for the zone containing the Top Level Domain (TLD), in this case, the .com TLD.
3. Once the DNS recursor receives the address of the TLD DNS server, it queries it for the address of the authoritative name-server for the megacorpone.com domain. The authoritative name-server is the final step in the DNS lookup process and contains the DNS records in a local database known as the zone file. It typically hosts two zones for each domain, the forward lookup zone that is used to find the IP address of a specific hostname and the reverse lookup zone (if configured by the administrator), which is used to find the hostname of a specific IP address.
4. Once the DNS recursor provides the DNS client with the IP address for www.megacorpone.com, the browser can contact the correct web server at its IP address and load the webpage.

DNS Enumeration

Interacting with a DNS Server

Each domain can use different types of DNS records. Some of the most common types of DNS records include:

- NS – Name-server records contain the name of the authoritative servers hosting the DNS records for a domain.
- A - Also known as a host record, the “a record” contains the IP address of a hostname (such as www.megacorpone.com).
- MX - Mail Exchange records contain the names of the servers responsible for handling email for the domain. A domain can contain multiple MX records.
- PTR - Pointer Records are used in reverse lookup zones and are used to find the records associated with an IP address.
- CNAME - Canonical Name Records are used to create aliases for other host records.
- TXT - Text records can contain any arbitrary data and can be used for various purposes, such as domain ownership verification.

```
kali@kali:~$ host -t ns megacorpone.com
```

```
C:\Users\administrator> nslookup
```

DNS Enumeration

Automating Lookups

Try Python with socket module

```
kali@kali:~$ python
>>> import socket
>>> socket.gethostbyname("notExistDomain.bkacad")
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
socket.gaierror: [Errno -5] No address associated with hostname
>>> socket.gethostbyname("www.megacorpone.com")
'3.220.87.155'
>>>
```

DNS Enumeration

Exercise

Write a small script to brute-force DNS domain from Megacorp-One. List of domain need to brute-force as: www, ftp, mail, owa, proxy, router.

Hint: `python, import socket, for, try except, List…`

DNS Enumeration

DNS Zone Transfer

A zone transfer is basically a database replication between related DNS servers in which the zone file is copied from a master DNS server to a slave server. The zone file contains a list of all the DNS names configured for that zone. Zone transfers should only be allowed to authorized slave DNS servers but many administrators misconfigure their DNS servers, and in these cases, anyone asking for a copy of the DNS server zone will usually receive one.

```
kali@kali:~$ host -l <domain name> <dns server address>
```

DNS Enumeration

DNSRecon

DNSRecon is an advanced, modern DNS enumeration script written in Python. Running `dnsrecon` against megacorpone.com using the `-d` option to specify a domain name, and `-t` to specify the type of enumeration to perform (in this case a zone transfer)

```
kali@kali:~$ dnsrecon -d megacorpone.com -t axfr
```

DNS Enumeration

DNSRecon

To begin the brute force attempt, we will use the `-d` option to specify a domain name, `-D` to specify a file name containing potential subdomain strings, and `-t` to specify the type of enumeration to perform (in this case `brt` for brute force)

```
kali@kali:~$ dnsrecon -d megacorpone.com -D ~/list.txt
-t brt
```

DNS Enumeration

Exercise

1. Recreate the example above and use `dnsrecon` to attempt a zone transfer from megacorpone.com.
2. **[Very-Hard]** Write a small script to attempt a zone transfer from megacorpone.com using a higher-level scripting language such as Python.

Hint: `import dns, import argparse, import socket, for, try except, List…`

DNS Enumeration

Exercise – Walkthrough

```python
#!/usr/bin/env python

import argparse
import dns.zone
import dns.resolver
import socket

def main(address):
    soa_answer = dns.resolver.query(address, 'SOA')
    master_answer = dns.resolver.query(soa_answer[0].mname, 'A')
    try:
        z = dns.zone.from_xfr(dns.query.xfr(master_answer[0].address, address))
        names = z.nodes.keys()
        names.sort()
        for n in names:
            print(z[n].to_text(n))
    except socket.error as e:
        print('Failed to perform zone transfer:', e)
    except dns.exception.FormError as e:
        print('Failed to perform zone transfer:', e)


if __name__ == '__main__':
    parser = argparse.ArgumentParser(description='DNS Python')
    parser.add_argument('--address', action="store", dest="address",  default='dnspython.org')
    given_args = parser.parse_args()
    address = given_args.address
    main (address)
```