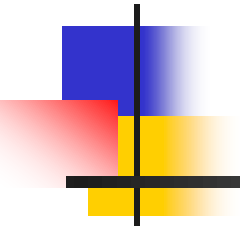
Penetration Testing



Lecture 08

Instructor: Dr. Cong Pu, Ph.D.

cong.pu@okstate.edu

Acknowledgment: Adapted partially from course materials from Dr. Wenliang Du at Syracuse University, Dr. Fengwei Zhang at Southern University of Science and Technology, and Dr. Steven M. Bellovin at Columbia University.

Penetration Testing

- Penetration Testing or Pen Testing:
  - a "*simulated*" ***cyberattack*** on a computer system/application
    - goals:
      1. identify *vulnerabilities* that could be exploited by malicious hackers
      2. uncover *security weaknesses* and provide recommendations to improve the system's security
  - also called: black box testing or ethical hacking
    - use automated *tools* and *manual techniques* to mimic the actions of real attackers

Penetration Testing (cont.)

- Pen Testing: the art of "*attacking*" a running application to find security vulnerabilities
  - *without knowing* the inner workings of the application itself
  - have access to an application as if they were users
    - given a valid account on the system
    - acts like an attacker and attempts to find and exploit vulnerabilities
- Pen testing helps
  - organizations understand how their systems can be compromised
  - allows organizations to take proactive measures to protect their data and assets

Penetration Testing (cont.)

- Advantages:
  - identifying vulnerabilities
  - improving security posture
  - can be fast (and therefore cheap)
  - requires a relatively lower skill-set than source code review
  - tests the code that is actually being exposed

- Disadvantages:
  - too late in the SDLC
  - front impact testing only
  - limited scope

Conduct Search Engine Discovery for Information Leakage: Summary

- Direct and indirect elements to search engine discovery
  - direct methods: searching the indexes and the associated content from caches
  - indirect methods: gleaning sensitive design and configuration information by searching forums, newsgroups, and etc.

Conduct Search Engine Discovery for Information Leakage: Test Objectives

- To understand
  - what *sensitive design and configuration information* of the application/system/organization is exposed both directly (on the organization's website) or indirectly (on a third party website)

Conduct Search Engine Discovery for Information Leakage: How to Test

- Use a search engine to search for:
  - Network diagrams and configurations
  - Archived posts and emails by administrators and other key staff
  - Log on procedures and username formats
  - Usernames and passwords
  - Error message content
  - Development, test, UAT and staging versions of the website

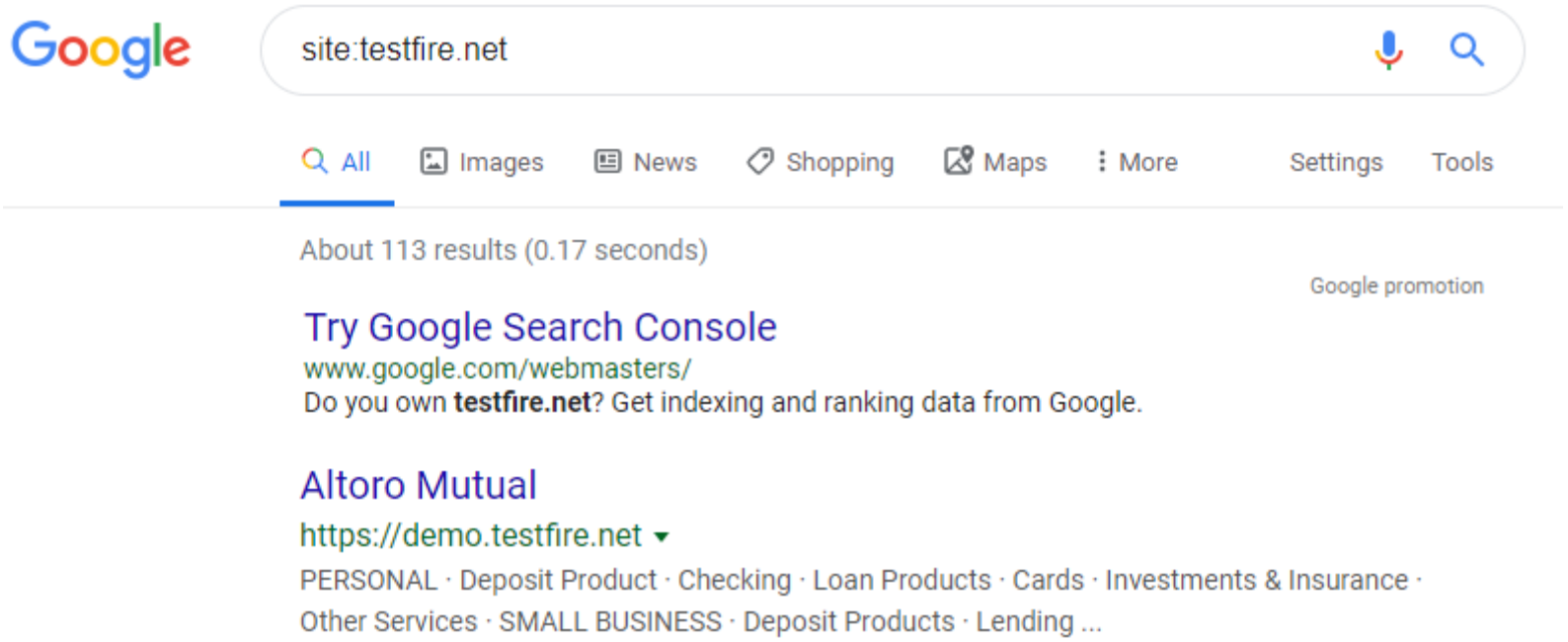
Conduct Search Engine Discovery for Information Leakage: Search Operators

- Using the advanced search operator: "site:"
  - restrict search results to a specific domain
- Do not limit testing to just one search engine provider
  - search engines: Baidu, Bing, Duck Duck Go, Startpage, Google, Shodan, PunkSpider
  - generate different results depending on when they crawled content and their own algorithms

Conduct Search Engine Discovery for Information Leakage: Example

- To find the web content of testfire.net indexed by a typical search engine, the syntax required is:

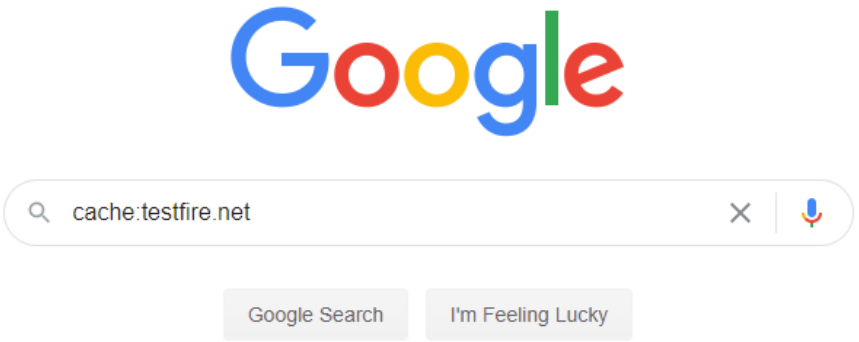
site: testfire.net



Conduct Search Engine Discovery for Information Leakage: Example

- To display the index.html of testfire.net as cached, the syntax is:

cache: testfire.net



This is Google's cache of https://demo.testfire.net/. It is a snapshot of the page as it appeared on Dec 20, 2019 11:08:47 GMT. The current page could have changed in the meantime. Learn more.

Full version Text-only version View source

Tip: To quickly find your search term on this page, press Ctrl+F or ⌘-F (Mac) and use the find bar.

Fingerprint Web Server: Summary

- Web server fingerprinting: a critical task for penetration testing
  - identifying the *version* and *type* of a running web server
    - determine known vulnerabilities and the appropriate exploits to use during testing
- Several different vendors and versions of web servers on the market today
  - knowing the type of web server that is being tested significantly helps in the testing process

Fingerprint Web Server: Summary

- Sending the web server specific commands and analyzing the output
  - each version of web server software may respond differently to these commands
  - knowing how each type of web server responds to specific commands

Fingerprint Web Server: Test Objectives

- Find the version and type of a running web server
  - determine known vulnerabilities and the appropriate exploits to use during testing

Fingerprint Web Server: How to Test

- Simplest and most basic form of identifying a web server: check the Server field in the HTTP response header
- Netcat
  - a networking utility which reads and writes data across network connections, using the TCP/IP protocol
  - a reliable "back-end" tool that can be used directly or easily driven by other programs and scripts
  - a feature-rich network debugging and exploration tool
    - create almost any kind of connection you would need and has several interesting built-in capabilities

Fingerprint Web Server: Netcat

- We will use the following syntax for nc command.

netcat options destination port

  - options used to set some special behavior like timeout, help, etc.
  - destination is used to specify remote system IP or Hostname
  - port is the remote system port number

Fingerprint Web Server: Netcat

- Help

netcat -h

- netcat command provides a lot or different potions

Fingerprint Web Server: Netcat

- Port Scan

netcat -z -v destination port#_range

  - -z option: zero-I/O mode
  - -v option: detailed information

Fingerprint Web Server: Example

- Look at the Server field in the HTTP response header to identify a web server

```
root@kali:~# nc testfire.net 80
HEAD / HTTP/1.0

HTTP/1.1 200 OK
Server: Apache-Coyote/1.1
Set-Cookie: JSESSIONID=9D783EDD4942C2463C18CAF1092DABE8; Path=/; HttpOnly
Content-Type: text/html;charset=ISO-8859-1
Date: Sat, 21 Dec 2019 21:00:52 GMT
Connection: close

root@kali:~#
```

Fingerprint Web Server: Example

- Look at the Server field in the HTTP response header to identify a web server

```
root@kali:~# nc testfire.net 80
GET / HTTP/3.0

HTTP/1.1 505 HTTP Version Not Supported
Server: Apache-Coyote/1.1
Date: Sat, 21 Dec 2019 21:03:07 GMT
Connection: close

root@kali:~#
```

Fingerprint Web Server: Example

- Look at the Server field in the HTTP response header to identify a web server

```
root@kali:~# nc testfire.net 80
GET / HTTP/1.0

HTTP/1.1 200 OK
Server: Apache-Coyote/1.1
Set-Cookie: JSESSIONID=8C66E7204E09212B5BF9D6A5CDA4C476; Path=/; HttpOnly
Content-Type: text/html;charset=ISO-8859-1
Date: Sat, 21 Dec 2019 21:04:03 GMT
Connection: close

<!-- BEGIN HEADER -->
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/
DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" >
```

Enumerate Applications on Webserver: Summary

- A paramount step in testing for web application vulnerabilities: find out which particular applications are hosted on a web server
  - many applications have known vulnerabilities and known attack strategies that can be exploited
    - gain remote control or to exploit data
  - many applications are often misconfigured or not updated, due to
    - the perception that they are only used "internally" and therefore no threat exists

Enumerate Applications on Webserver: Test Objectives

- Enumerate the applications within scope that exist on a web server

Enumerate Applications on Webserver: How to Test

- While web applications usually live on port 80 (http) and 443 (https)
  - in fact, web applications may be associated with arbitrary TCP ports, and can be referenced by specifying the port number as follows: http[s]://www.example.com:port/.
    - e.g., http://www.example.com:20000/

Enumerate Applications on Webserver: How to Test

- Check for the existence of web applications on non-standard ports.
- A port scanner such as nmap is capable of performing service recognition
  - -sV option
    - identify http[s] services on arbitrary ports

Enumerate Applications on Webserver: Example

- Check for the existence of web applications on non-standard ports.
- A port scanner such as nmap is capable of performing service recognition by means of the -sV option, and will identify http[s] services on arbitrary ports.

```
root@kali:~# nmap -sV testfire.net
Starting Nmap 7.80 ( https://nmap.org ) at 2019-12-21 16:20 EST
Nmap scan report for testfire.net (65.61.137.117)
Host is up (0.015s latency).
Not shown: 997 filtered ports
PORT     STATE SERVICE    VERSION
80/tcp   open  http       Apache Tomcat/Coyote JSP engine 1.1
443/tcp  open  ssl/https?
8080/tcp open  http       Apache Tomcat/Coyote JSP engine 1.1

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 73.74 seconds
root@kali:~#
```