



# **Module 3: Vulnerability Discovery**

©2010 C-DAC, Hyderabad

# Objective

- In this module we are going to discuss about the topics
  - What is Vulnerability?
  - Different types of Vulnerabilities
    - » Web Server Vulnerabilities
    - » Database Vulnerabilities
    - » Common Vulnerabilities
  - Anatomy of Vulnerability Discovery
  - What is Vulnerability Discovery?
  - Vulnerability Management
  - Finding Vulnerabilities in the Web Application using scanning tools like
    - » GooLag, Wikto, Nikto, Nessus, Nmap, Burpsuite

©2010 C-DAC, Hyderabad

In this module we would cover the topics related to Vulnerabilities and Web Application Vulnerabilities how to discover and mitigate them. In each topic we will cover in depth how the attackers are working to find the Vulnerabilities on the Web Server as well as Web Application. We will also discuss the attack sophisticated graph, the increase in attacks using different techniques and tools. We will also discuss about the Web Vulnerability scanning tools for finding different types of Vulnerabilities present on the Web Server and Web Application and the steps or countermeasures to be followed to mitigate the risks.



# What is Vulnerability?

©2010 C-DAC, Hyderabad

Vulnerability is a potential avenue of attack. Vulnerabilities may exist in computer systems and networks (allowing the system to be open to a technical attack) or in administrative procedures (allowing the environment to be open to a non-technical or social engineering attack).

When identifying vulnerabilities, begin by locating all the entry points to the organization. In other words, find all the access points of information (in both electronic and physical form) and systems within the organization. This means identifying the Internet connections, remote access points, connections to other organizations, physical access to facilities, user access points and wireless access points.

For each of these access points, identify the information and systems that are accessible. Then identify how the information and systems may be accessed. Be sure to include in this list any known vulnerabilities in operating systems and applications.

Vulnerabilities are security flaws in OS's, networks, cross platform applications, usually caused by poorly written and insecure code.



## Different types of Vulnerabilities

- Web Server Vulnerabilities
- Database Vulnerabilities
- Common vulnerabilities

©2010 C-DAC, Hyderabad

There are different kinds of Vulnerabilities present on the Web Server or Web Application. They are Web Server, Database and common Vulnerabilities. Web Server Vulnerabilities are the Vulnerabilities present on the Web Servers like Apache Web Server mostly called as HTTP Web Server, Apache tomcat, Internet Information Services (IIS), Sun java Web Server, lighttpd, jigsaw, kclone, etc..... The main reason behind this is the Server maintenance people are not upto date with the latest patches and not taking secure precautions for closing their Server running ports. Database Vulnerabilities are the Vulnerabilities present in the database Software like MySQL, MsSQL...etc and incorrect configuration of databases. Common Vulnerabilities can be anything like weak passwords, allowing scripts in the applications...etc.

In depth description of each vulnerability is discussed in future slides.



# Web Server Vulnerabilities

- All popular web sites
- Part of Internet
- OS vulnerabilities like misconfigurations
- Easier to exploit because of poor security

©2010 C-DAC, Hyderabad

Web Servers and web-based applications are popular targets for attacks. Web Servers are usually available through corporate firewalls, and web-based applications are often developed without following a thorough security method. To detect web-based attacks, Intrusion Detection Systems (IDS) are organized with a number of signatures that support the identification of known attacks. As huge amount of vulnerabilities are discovered daily keeping the Intrusion Detection Systems (IDS) updated is hard. Additionally, vulnerabilities may be introduced by custom web-based applications developed internally.

All popular websites which are part of Internet are prone to vulnerabilities. Many web sites have vulnerabilities like misconfigurations in operating systems, web applications. And some web sites can be exploited because of poor security.



# Database Vulnerabilities

- Major Databases
- Support to Web Servers
- Reside in DMZ along with Web Servers
- OS vulnerabilities
- TCP communication vulnerabilities

©2010 C-DAC, Hyderabad

## **Database Vulnerabilities:**

Vulnerabilities are like holes that are used to crack into the database. A database acts as backend to many web sites and support for Web Servers where the data is stored in the database. Database servers are traditionally protected using firewalls and are located inside De-Militarized Zones (DMZ). Majority of the databases have vulnerabilities. They may be operating system vulnerabilities or TCP communication vulnerabilities.

There is a need to run vulnerability scans on databases to find misconfigurations and vulnerabilities in the database software before launching the Web Site to public.



## Cont....

- Default passwords for Database(DB)s
- Configurations of databases
  - Permissions, Roles
- Constructions of DB objects
  - Tables, DML, DDL statements
- Injection Attacks
- Application attacks
  - Poorly written for Databases

©2010 C-DAC, Hyderabad

The common vulnerabilities that exist in the databases are due to improper configuration of databases, granting permissions and roles to the objects in database, and construction of Database objects like database tables, commands executed etc. Default passwords for databases need to be changed. And attacks like SQL Injection and application attacks make use of vulnerabilities like poorly written databases.



# Common vulnerabilities

- Passwords
- Traverse Directories
- Scripts and Programs
- Bypassing URLs
- Patches

©2010 C-DAC, Hyderabad

Some of the common vulnerabilities that may occur in network level, application level and system level are:

Passwords can be cracked using vulnerabilities in applications or by using the password cracking methods like Dictionary attacks, Brute force methods, etc,. Password cracking is the method of recovering passwords from database that has been stored in. A common way is to repeatedly try guesses for the password. The use of password cracking may help the user to recover a lost password or to gain unauthorized access to a system or Web site.

The vulnerabilities in programs can be exploited using scripts. Scripts can be used to insert malware (malicious software). One of the methods of scripting attack is Cross site scripting. Cross site scripting means malicious script is executed in Client's Web Browser. By using these scripts, attacker can act as administrator. The attacker can steal user's accounts Credentials, can modify the Web pages and also can execute any command at the client machine.

Even the patches that are released to stop vulnerabilities in software may also introduce a new type of vulnerabilities. The delay of releasing the patches also might cause problems.





## Cont....

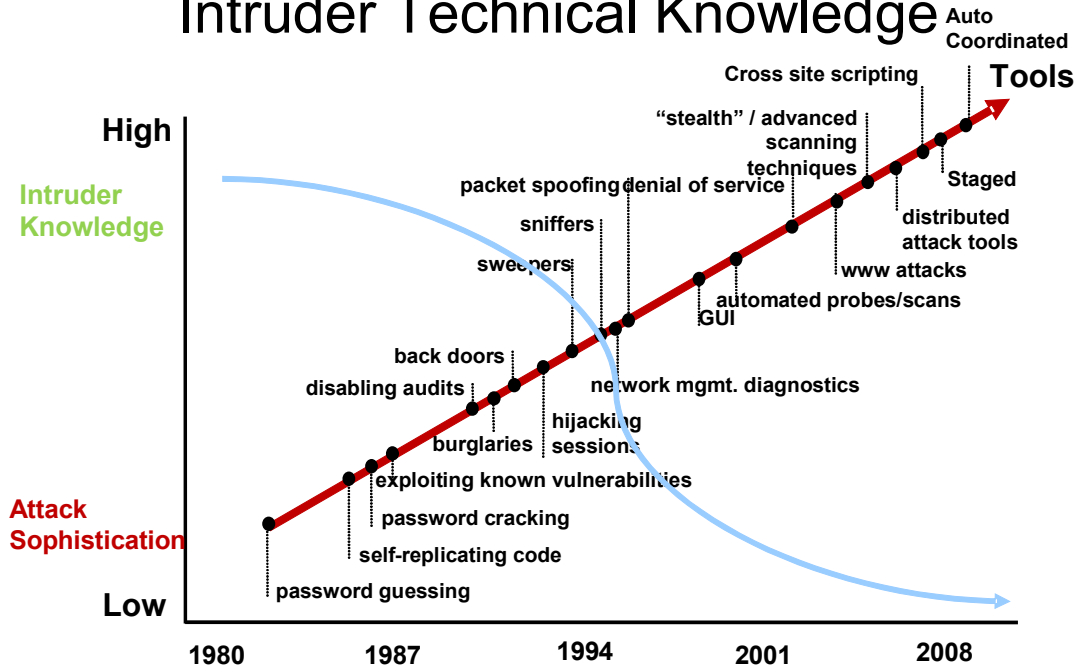
- Root access
- Web site defacement
- Full Database control
- Denial of Service
- Theft or alternation of data
- More penetration

©2010 C-DAC, Hyderabad

An attacker may use the Web Server vulnerabilities to gain system privileges like Root access in Unix or Linux systems and Administrator in Windows systems. With the highest system privilege access, the attacker gets full control on the Web Server. Using the vulnerabilities, web site defacement can be done. There may be theft of web site data or data in web site can be modified. Denial of Service attack blocks the use of system resources or web site resources to the legitimate users.



# Attack Sophistication vs. Intruder Technical Knowledge

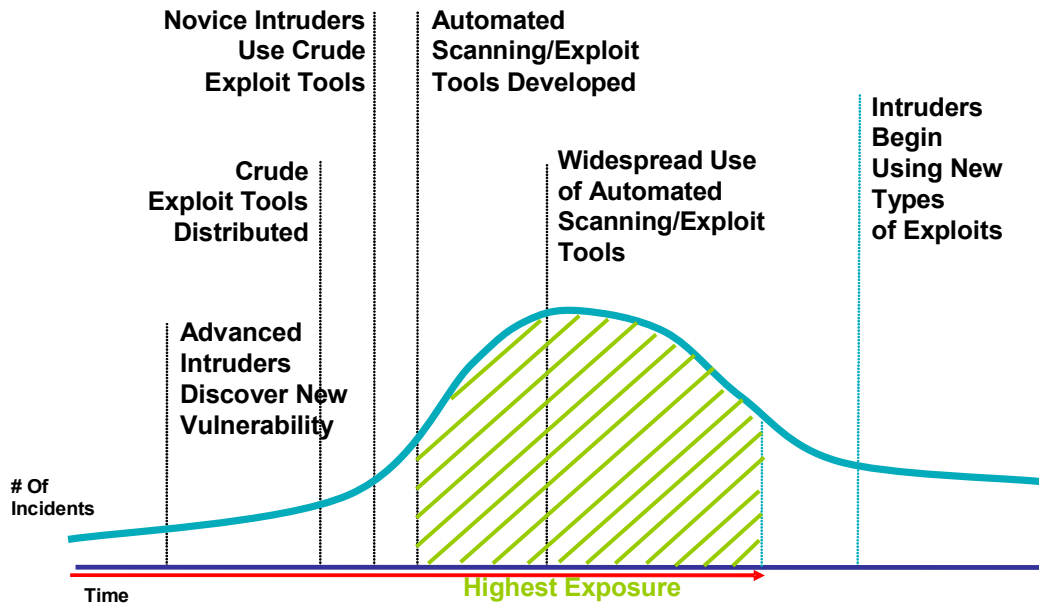


©2010 C-DAC, Hyderabad

The above graph shows the Attack Sophistication vs. Intruder Technical Knowledge. From the graph we can understand that the attackers have become very clever and used different new techniques for performing different kinds of cyber attacks. If we observe the graph from 1980 to 2008 the attack scenario is increased drastically and we can also observe that the usage of tools is also increased than using human knowledge. The software tool designers have become very intelligent in writing the code, which performs a dangerous or exploitable action in the victim's system. In day to day life the intruders are becoming very low and the attack tools usage is becoming very high.



# Vulnerability Exploit Cycle



©2010 C-DAC, Hyderabad

The above graph shows the Vulnerability Exploit Cycle. From the graph we can observe that the intruders are finding the vulnerabilities in the victim system by performing different network scans and then performing the attacks by using most advanced exploitable tools.



# Vulnerability Scanners

- The concept is advanced to port scanner
- Extracting the information from hosts
- Known vulnerabilities from public sources
- Performing tests against them
- Rating the risk of vulnerability
- Providing fixing of direction
- Report generation

©2010 C-DAC, Hyderabad

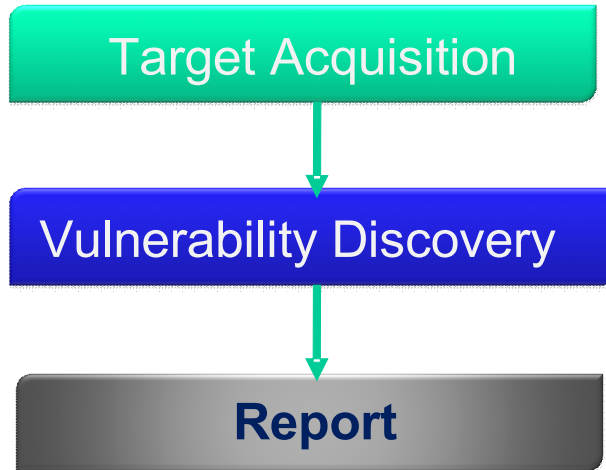
Vulnerability Discovery takes the concept of a port scanner. It not only identifies hosts and open ports and all associated vulnerabilities automatically.

Automated Network vulnerability scanners extract information from the target hosts like OS version, open ports, active services and protocols, version of each running service, exported resources and shares, valid accounts by checking all details against publicly available sources of known vulnerability information and vendor security alerts to see if the known potential vulnerabilities are present.

Performs tests and confirms the existence of real vulnerability and rates the risk of vulnerability. It also does mapping of each finding to their related security alert with fixing direction of vulnerability. Finally it creates a report.



# Anatomy



©2010 C-DAC, Hyderabad

Vulnerability Discovery has lot of things in general with risk assessment. Evaluations are typically performed according to the following steps:

- ☐ Listing of assets and capabilities (resources) in a system.
- ☐ Allocating resources in rank order and importance to those resources
- ☐ Discovering the vulnerabilities or potential threats to each resource
- ☐ Eliminating a good number of serious vulnerabilities for the important resources



# What is Vulnerability Discovery ?

©2010 C-DAC, Hyderabad

## **Vulnerability Discovery (VD):**

Vulnerability Discovery is the process of scanning and pointing out vulnerabilities in a Web application of the Server. It is the process of recognizing, quantifying, and prioritizing the vulnerabilities in an application. It is a systematic approach of evaluation of an organization's IT weaknesses of infrastructure components and assets and how those weaknesses can be mitigated through proper security controls and recommendations to remediate exposure to risks, threats, and vulnerabilities.

Vulnerability discovery is a method or a process which is used to reveal and fix types of software defects with security impacts (impacts like crash or collision) when present in information systems.

During the process of producing software products, the software engineers unexpectedly or unintentionally creates vulnerabilities, which are later discovered and removed or mitigated. Software designers think that they are paying greater attention to all the phases of the development lifecycle and avoid vulnerabilities.



## Cont...

- Part of ethical hacking / Penetration Testing
- Determines susceptibility
- Compile on inventory system
- Weakness finding
- Advantages of attackers techniques

©2010 C-DAC, Hyderabad

Vulnerability Discovery is treated as part of Ethical Hacking and Penetration Testing. A Vulnerability Discovery is being proactive. It determines one's susceptibility to an attack before networks are exploited, and it forces companies to take early corrective action. It can show the consequences of an attack to your organization. A Vulnerability Discovery is the procedure of identifying, quantifying, and prioritizing (or ranking) the vulnerabilities in a system.

The purpose of Vulnerability Discovery is to compile an inventory system and services attached to network system and for each system and service, identify the weakness and vulnerabilities visible, also taking advantages of attacker's techniques.



# Vulnerability Management

- Method of
  - ✓ Finding
  - ✓ Assessing
  - ✓ and remediating vulnerabilities
- Use Tools
  - Nessus
  - Nmap
  - WebScarab
  - Winkto
  - Nikto
- Regularly update the tools

©2010 C-DAC, Hyderabad

The **Vulnerability management** is the method or process of finding, assessing and remediating vulnerabilities (existing exploitable weaknesses) on servers and workstations. is the process which can be implemented in organizations to make IT environments more secure and to improve an organization's regulatory compliance posture.

The vulnerability management process includes the following steps:

- The **Policy** definition is the first step which includes the definition of the desired state for device configurations, user identity and resource access.
- The **Baseline** environment is used to identify the vulnerabilities and policy compliance.
- **Prioritize** the mitigation activities based on external threat information, internal security posture and asset classification.
- The **Shield** environment should be constructed, to eliminate the vulnerabilities by using the desktop and network security tools.
- After identifying the Vulnerabilities, **mitigate** the vulnerabilities and eliminate the root causes.
- **Maintain** the same process frequently and continually **monitor** the environment for deviations from policy and to identify new vulnerabilities.
- The technology which was provided by the vulnerability management vendors can be used to automate the various aspects of the vulnerability management process.



The four main technology categories are:

- 1. Vulnerability Discovery**
- 2. Security configuration management and policy compliance**
- 3. IT security risk management**
- 4. Security information and event management (SIEM)**

Use vulnerability scanning tools for performing vulnerability check on the Web Application. Some of the tools are mentioned below.

Nessus: Nessus Security Scanner

Nmap: Security Scanner

WebScarab: Framework for analyzing the Applications

Winkto: Web Server Assessment tool

Nikto: Web Scanner



# Web Application Vulnerability Detection

- Detect all kind of vulnerabilities
  - SQL injection
  - XSS
  - Xpath injection
  - CSRF
  - Form weak password

©2010 C-DAC, Hyderabad

To detect Vulnerability in the Web Application:

1. An In-depth Scan on the web application should be performed (Need to check is there any possibility of accessing the back-end database information and website list).
2. Detect all kinds of web vulnerabilities deeply such as Cross Site Scripting, Cross Site Request Forgery, SQL injection, Xpath injection, the form around, form default password, blank password, weak password, all kinds of CGL vulnerabilities.
3. Detect is there any possibility of Web Trojans.
4. Perform complete penetration testing on the target Web Application by implementing sound attack to obtain direct evidence of system security threats by imitating the vulnerability discovery techniques and attack methods of the hacker to current vulnerability.
5. Perform configuration Audit by implementing database auditing capabilities and access to connection information in back-end database, database instance name, database version, data dictionary, and some other configuration information by imitating a hacker attack through current vulnerability.



# Web Application Vulnerability Detection using Scanners

- GooLag Scanner
- Wikto
- Nikto
- Nessus
- Nmap
- Burpsuite
- WebScarab
- Paros Proxy

©2010 C-DAC, Hyderabad

Web vulnerability scanning is a security development technology that sets website owners on the offensive in identifying and closing their website security vulnerabilities. Web vulnerability scanning generates clarifications and actionable steps on how to close the vulnerabilities.

Some of the Web Vulnerability Scanning tools are GooLag Scanner, Wikto, Nikto, WebScarab, Burpsuite, Paros Proxy. We will discuss each tool in detail in future slides.



# Goolag Scanner

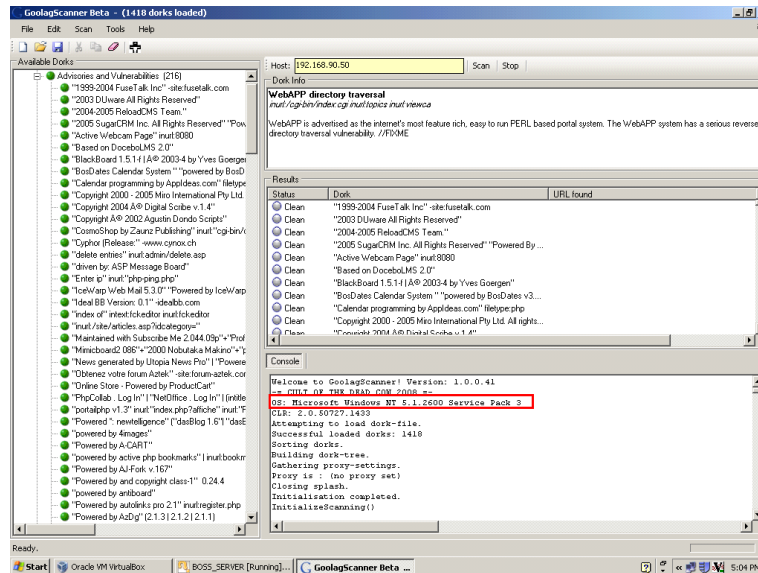
## Goolag Scanner

- » Website audit tool
- » The technology is based on Google Hacking
- » Developed by Johnny I hack stuff

©2010 C-DAC, Hyderabad

The Goolag Scanner is a Website Vulnerability scanning tool. It is an open source tool. It works by exploiting data- retention practices of popular search engines. The scanning technology is based on Google Hacking which is developed by **“Johnny I hack stuff”**.

# GooLag Scanner



©2010 C-DAC, Hyderabad

Download GooLag scanner tool from Internet and Scan the GoolagScanner executable file with anti-virus software and install it in the system by following the steps given by it. After installing the software open it by going to the location Start → All Programs → GooLagScanner → GooLagScanner. The screen will be displayed as shown in the figure. In the HOST column enter the IP address of the machine you want to scan and select one of the Dork which is at lefthand side and click on scan button. After scanning the system it will display the report as shown in the above figure.

From the figure we can observe that, in the left panel there are number of Available dorks used for scanning the Website. And in right hand side panel we can find the Web Application Directory traversal, Results of the scanning and at the bottom we can find the report console.

The continuation output is mentioned below

### The tool has used the following Dorks for scanning the Website

"1999-2004 FuseTalk Inc" -site:fusetalk.com

"2003 DUware All Rights Reserved"

"2004-2005 ReloadCMS Team."

"2005 SugarCRM Inc. All Rights Reserved" "Powered By SugarCRM"

"Active Webcam Page" inurl:8080

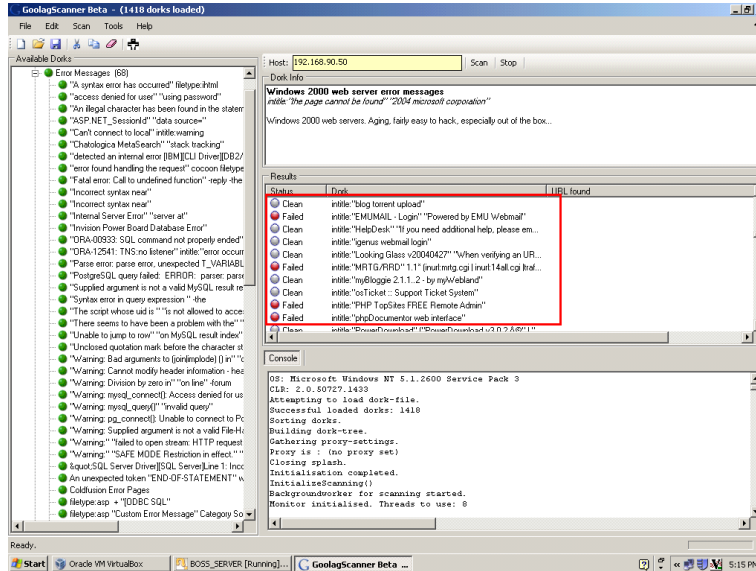
"Based on DoceboLMS 2.0"  
 "BlackBoard 1.5.1-f | Â© 2003-4 by Yves Goergen"  
 "BosDates Calendar System " "powered by BosDates v3.2 by BosDev"  
 "Calendar programming by AppIdeas.com" filetype:php  
 "Copyright 2000 - 2005 Miro International Pty Ltd. All rights reserved" "Mambo is Free Software released"  
 "Copyright 2004 Â© Digital Scribe v.1.4"  
 "Copyright Â© 2002 Agustin Dondo Scripts"  
 "CosmoShop by Zaunz Publishing" inurl:"cgi-bin/cosmoshop/lshop.cgi" -johnny.ihackstuff.com -V8.10.106 -V8.10.100 -V.8.10.85 -V8.10.108 -V8.11\*  
  
 "Cyphor (Release:" -www.cynox.ch  
 "delete entries" inurl:admin/delete.asp  
 "driven by: ASP Message Board"  
 "Enter ip" inurl:"php-ping.php"  
 "IceWarp Web Mail 5.3.0" "Powered by IceWarp"  
 "Ideal BB Version: 0.1" -idealbb.com  
 "index of" intext:fckeditor inurl:fckeditor  
 "inurl:/site/articles.asp?idcategory="   
 "Maintained with Subscribe Me 2.044.09p"+"Professional" inurl:".s.pl"  
 "Mimicboard2 086"+"2000 Nobutaka Makino"+"password"+"message" inurl:page=1  
  
 "News generated by Utopia News Pro" | "Powered By: Utopia News Pro"  
 "Obtenez votre forum Aztek" -site:forum-aztek.com  
 "Online Store - Powered by ProductCart"  
 "PhpCollab . Log In" | "NetOffice . Log In" | (intitle:"index.of." intitle:phpcollab|netoffice inurl:phpcollab|netoffice -gentoo)  
 "portailphp v1.3" inurl:"index.php?affiche" inurl:"PortailPHP" -site:safari-msi.com  
  
 "Powered \*: newtelligence" ("dasBlog 1.6"| "dasBlog 1.5"| "dasBlog 1.4"| "dasBlog 1.3")  
  
 "powered by 4images"  
 "Powered by A-CART"  
 "powered by active php bookmarks" | inurl:bookmarks/view\_group.php?id=  
  
 "Powered by AJ-Fork v.167"  
 "Powered by and copyright class-1" 0.24.4  
 "powered by antiboard"  
 "Powered by autolinks pro 2.1" inurl:register.php  
 "Powered by AzDg" (2.1.3 | 2.1.2 | 2.1.1)  
 intext:"LinPHA Version" intext:"Have fun"  
 intext:"PhpGedView Version" intext:"final - index" -inurl:demo  
 intext:"Powered by CubeCart 3.0.6" intitle:"Powered by CubeCart"  
 intext:"Powered by DEV web management system" -dev-wms.sourceforge.net -demo  
  
 inurl: "/cgi-bin/loadpage.cgi?user\_id="

inurl:"/login.asp?folder=" "Powered by: i-Gallery 3.3"  
inurl:"/site/articles.asp?idcategory="  
inurl:"comment.php?serendipity"  
inurl:"extras/update.php" intext:mysql.php -display  
inurl:"forumdisplay.php" +"Powered by: vBulletin Version 3.0.0..4"

### **Scanned Output**

Welcome to GoolagScanner! Version: 1.0.0.41  
- = CULT OF THE DEAD COW 2008 = -  
OS: Microsoft Windows NT 5.1.2600 Service Pack 3  
CLR: 2.0.50727.3053  
Attempting to load dork-file.  
Successful loaded dorks: 1418  
Sorting dorks.  
Building dork-tree.  
Gathering proxy-settings.  
Proxy is : (no proxy set)  
Closing splash.  
Initialisation completed.  
Mass scan, count of dorks exceeds: 10  
InitializeScanning()  
Backgroundworker for scanning started.  
Monitor initialised. Threads to use: 8  
Monitor: Watcher started.  
ScanURL:: http://www.google.com/search?q=%221999-2004%20FuseTalk%20Inc%22%20+site:192.168.90.50  
ScanURL:: http://www.google.com/search?q=%222003%20DUware%20All%20Rights%20Reserved%22+site:192.168.90.50  
Count of results : 0  
Next page to request : 0  
ScanURL:: http://www.google.com/search?q=%222004-2005%20ReloadCMS%20Team.%22+site:192.168.90.50  
  
http://www.google.com/search?q=%222005%20SugarCRM%20Inc.%20All%20Rights%20Reserved%22%20%22Powered%20By%20SugarCRM%22+site:192.168.90.50  
http://www.google.com/search?q=intitle:guestbook%20inurl:guestbook%20%22powered

# GooLag Scanner



©2010 C-DAC, Hyderabad

From the figure we can observe that the scanned status of the host 192.168.90.50.

## Findings:

The above screen shot is the output of the scanned host i.e 192.168.90.50. Here we can find the Operating system of the scanned host that is **OS: Microsoft Windows NT 5.1.2600** and the details of the service pack is **Service Pack 3**. And we can also find that the GooLag Scanner is using different kinds of google search engine techniques (site:, inurl, index of, intitle, intext..etc) to find the vulnerabilities in the website.





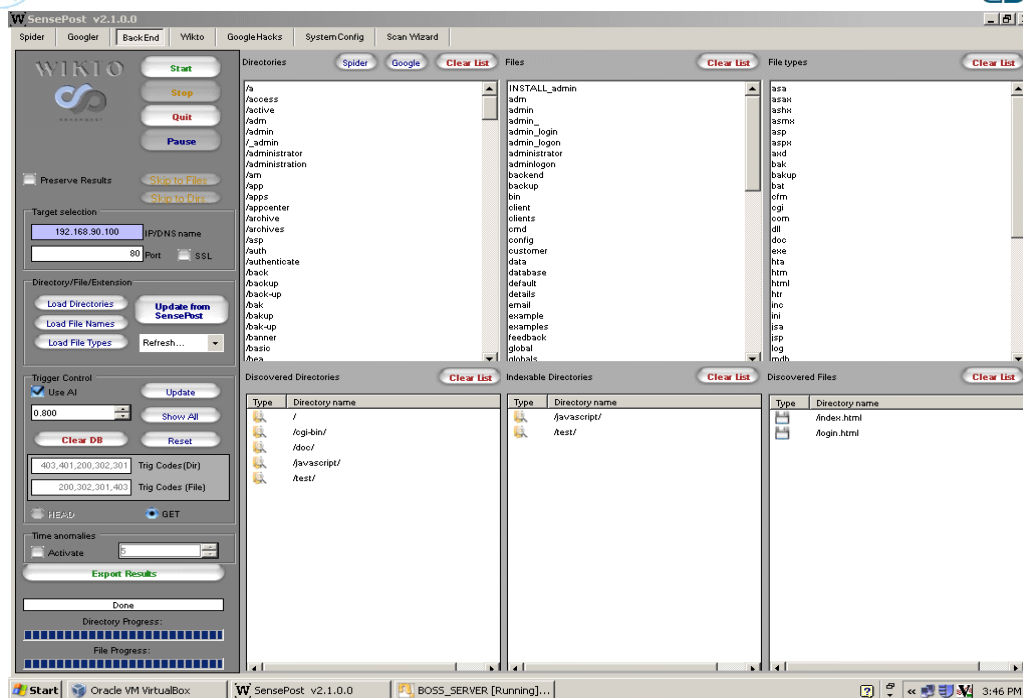
# Wikto

## Wikto

- » Web Vulnerability Scanner
- » Created by Sensepost
- » Freeware
- » Windows based tool

©2010 C-DAC, Hyderabad

Wikto is a Web Vulnerability Scanning tool, which checks for flaws in the Webservers. It is a freeware and is created by Sensepost. Its functions are same as Nikto, but Windows version. It has some of the functionality like Back-End miner and close Google integration.



©2010 C-DAC, Hyderabad

Download Wikto tool from <http://www.sensepost.com/labs/tools/pentest/wikto> and Scan the Wikto executable file with anti-virus software and install it in the system by following the steps given by it. After installing the software open it by going to the location Start → All Programs → SensePost → Wikto → Wikto. Go to Wikto tab or BackEnd tab and enter the IP address and the port number of the machine which you want to scan for finding the Web Vulnerabilities of that Web Server and click on the start button to start the scanning process.

The above figure is the scanned output of the host (WebServer) 192.168.90.100.

### Findings:

From the figure we can observe that it has scanned the entire host and displayed all the **directories** and the **files** existing on the Web Server. By using this scenario the attackers try to gather the valuable information of the Web server like what are the files hosted on the server.



# Nikto

## Nikto

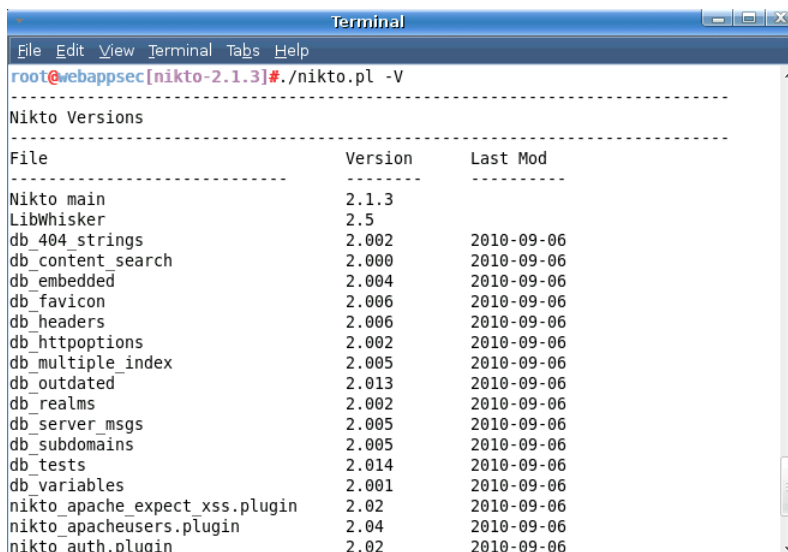
- » WebServer Scanner
- » Open Source tool
- » Written by Chris Sullo and David Lodge

©2010 C-DAC, Hyderabad

Nikto is a Web Vulnerability Scanner. It is an Open Source web server scanner tool, which performs a comprehensive test against the web servers for multiple items including over 6400 potentially dangerous files/CGIs. It checks for out of date versions of over 1000 servers, and version exact problems on over 270 servers. It also checks for server configuration items such as the existence of multiple index files, HTTP server options, and will attempt to identify installed web servers and software. It was written by Chris Sullo and David Lodge.



# Web Scanning using Nikto



```
Terminal
File Edit View Terminal Tabs Help
root@webappsec[nikto-2.1.3]#./nikto.pl -V
-----
Nikto Versions
-----
File                               Version    Last Mod
-----
Nikto main                         2.1.3
LibWhisker                         2.5
db_404_strings                     2.002      2010-09-06
db_content_search                  2.000      2010-09-06
db_embedded                        2.004      2010-09-06
db_favicon                         2.006      2010-09-06
db_headers                         2.006      2010-09-06
db_httptoptions                    2.002      2010-09-06
db_multiple_index                  2.005      2010-09-06
db_outdated                        2.013      2010-09-06
db_realms                          2.002      2010-09-06
db_server_msgs                     2.005      2010-09-06
db_subdomains                      2.005      2010-09-06
db_tests                           2.014      2010-09-06
db_variables                       2.001      2010-09-06
nikto_apache_expect_xss.plugin     2.02       2010-09-06
nikto_apacheusers.plugin           2.04       2010-09-06
nikto_auth.plugin                  2.02       2010-09-06
```

©2010 C-DAC, Hyderabad

## Web Scanning – Nikto: a powerful Web Scanning tool

Nikto is an Open Source (GPL) web server scanner which performs comprehensive tests against web servers for multiple items, including over 3500 potentially dangerous files/CGIs, versions on over 900 servers, and version specific problems on over 250 servers. Scan items and plugins are regularly updated and can be automatically updated (if desired). Nikto is not designed as an overly stealthy tool

Download Nikto tool from <http://cirt.net/nikto2> and install it in the system by following the steps given by it. After installing the software, open the terminal and execute the command as shown in the above figure to find nikto different versions.

### -V(Version):

Nikto version 2 contains many improvements over the first version. Some of the major new features include:

- Fingerprinting web servers via favicon.ico files

- 404 checking for each file type

- Improved false positive reduction via multiple methods: headers, page content, and content hashing

- Scan tuning to include or exclude entire classes of vulnerability tests

- Expanded scan database can have several positive or negative triggers, to allow AND/OR/NOT for flexible checks

A "single" scan mode that allows you to craft an HTTP request by hand  
 Updated and greatly enhanced documentation  
 Authorization guessing holds any directory, not just the root directory  
 New HTML report  
 The Basic pattern or template engine so that HTML reports can be easily customized  
 An experimental knowledge base for scans, which will allow renewed reports and retests  
 (future)

**Ex:**

**root@webappsec[nikto-2.1.3]#./nikto.pl -V**

---

#### Nikto Versions

---

File	Version	Last Mod
Nikto main	2.1.3	
LibWhisker	2.5	
db_404_strings	2.002	2010-09-06
db_content_search	2.000	2010-09-06
db_embedded	2.004	2010-09-06
db_favicon	2.006	2010-09-06
db_headers	2.006	2010-09-06
db_httptoptions	2.002	2010-09-06
db_multiple_index	2.005	2010-09-06
db_outdated	2.013	2010-09-06
db_realms	2.002	2010-09-06
db_server_msgs	2.005	2010-09-06
db_subdomains	2.005	2010-09-06
db_tests	2.014	2010-09-06
db_variables	2.001	2010-09-06
nikto_apache_expect_xss.plugin	2.02	2010-09-06
nikto_apacheusers.plugin	2.04	2010-09-06
nikto_auth.plugin	2.02	2010-09-06
nikto_cgi.plugin	2.04	2010-09-06
nikto_content_search.plugin	2.03	2010-09-06
nikto_core.plugin	2.1.4	2010-09-06
nikto_dictionary_attack.plugin	2.02	2010-09-06
nikto_embedded.plugin	2.05	2010-09-06
nikto_favicon.plugin	2.07	2010-09-06
nikto_headers.plugin	2.08	2010-09-06
nikto_httptoptions.plugin	2.08	2010-09-06
nikto_msgs.plugin	2.05	2010-09-06
nikto_multiple_index.plugin	2.01	2010-09-06
nikto_outdated.plugin	2.07	2010-09-06
nikto_put_del_test.plugin	2.03	2010-09-06
nikto_report_csv.plugin	2.04	2010-09-06
nikto_report_html.plugin	2.04	2010-09-06



**`./nikto.pl -host <webserver address>`**

```
Terminal
File Edit View Terminal Tabs Help
root@webappsec[nikto-2.1.3]# ./nikto.pl -h testserver.webappsec.com
- Nikto v2.1.3

+ Target IP: 192.168.90.50
+ Target Hostname: testserver.webappsec.com
+ Target Port: 80
+ Start Time: 2010-10-07 23:16:42

+ Server: Apache/2.2.11 (Win32) PHP/5.3.0
+ No CGI Directories found (use '-C all' to force check all possible dirs)
+ Apache/2.2.11 appears to be outdated (current is at least Apache/2.2.16). Apache 1.3.42 and 2.0.63 are also current.
+ PHP/5.3.0 appears to be outdated (current is at least 5.3.2)
+ OSVDB-877: HTTP TRACE method is active, suggesting the host is vulnerable to XST
+ OSVDB-3268: /icons/: Directory indexing found.
+ 6417 items checked: 1 error(s) and 4 item(s) reported on remote host
+ End Time: 2010-10-07 23:17:53 (71 seconds)

+ 1 host(s) tested
root@webappsec[nikto-2.1.3]#
```

©2010 C-DAC, Hyderabad

By using nikto tool we can scan the target by using the webserver address.

Let us see the example

#### **Command of nikto:**

**`./nikto.pl -h <webserver address>`**

Here, **nikto.pl** is the executable command.

**-h** is the option for specifying the target host name or IP address

**testserver.webappsec.com** is the website address of the target machine.

#### **Findings:**

From the output we can find that the details of the target machine like IP address (**192.168.90.50**), hostname (**testserver.webappsec.com**), port number (**80**) used by the webserver. We can also find the details like which webserver and the version they have used (**Apache 2.2.11**) and we can also find that the Webserver and the PHP used are outdated versions.

And there is vulnerable in the HTTP that the HTTP TRACE method is in active. By observing the Nikto output we need to close the loopholes which were found in the scanning.



**./nikto.pl –host <IP Address>**

```
root@webappsec[nikto-2.1.3]# ./nikto.pl -h 192.168.90.50
- Nikto v2.1.3

+ Target IP: 192.168.90.50
+ Target Hostname: testserver.webappsec.com
+ Target Port: 80
+ Start Time: 2010-10-07 23:17:03

+ Server: Apache/2.2.11 (Win32) PHP/5.3.0
+ No CGI Directories found (use '-C all' to force check all possible dirs)
+ Apache/2.2.11 appears to be outdated (current is at least Apache/2.2.16). Apache 1.3.42 and 2.0.63 are also current.
+ PHP/5.3.0 appears to be outdated (current is at least 5.3.2)
+ OSVDB-877: HTTP TRACE method is active, suggesting the host is vulnerable to XSS
+ OSVDB-3268: /icons/: Directory indexing found.
+ 6417 items checked: 1 error(s) and 4 item(s) reported on remote host
+ End Time: 2010-10-07 23:18:14 (71 seconds)

+ 1 host(s) tested
root@webappsec[nikto-2.1.3]#
```

©2010 C-DAC, Hyderabad

By using nikto tool we can also scan the target by using the IP address of it.

Let us see the example

#### **Command of nikto:**

**./nikto.pl –h <IP address>**

Here, **nikto.pl** is the executable command.

**-h** is the option for specifying the target host name or IP address

**192.168.90.50** is the website address of the target machine.

#### **Findings:**

From the output we can find that the details of the target machine like IP address, hostname, port number used by the webserver. We can also find the details like which webserver and the version they have used (Apache 2.2.11) and we can also find that the Webserver and the PHP used are outdated versions. And there is vulnerable in the HTTP that the HTTP TRACE method is in active.

With this information the attacker may find the bugs in the above mentioned apache web sever and php and take advantage of those bugs to make create his attack successful.



## `./nikto.pl -h <IP Address> -Cgidirs`

```
Terminal
File Edit View Terminal Tabs Help
root@webappsec[nikto-2.1.3]#./nikto.pl -h testserver.webappsec.com -Cgidirs all
- Nikto v2.1.3
-----
+ Target IP:      192.168.90.50
+ Target Hostname: testserver.webappsec.com
+ Target Port:    80
+ Start Time:     2010-10-07 21:07:58
-----
+ Server: Apache/2.2.11 (Win32) PHP/5.3.0
+ Apache/2.2.11 appears to be outdated (current is at least Apache/2.2.16). Apache 1.3.42 and 2.0.63 are also current.
+ PHP/5.3.0 appears to be outdated (current is at least 5.3.2)
+ OSVDB-877: HTTP TRACE method is active, suggesting the host is vulnerable to XSS
+ OSVDB-3268: /icons/: Directory indexing found.
+ 6417 items checked: 1 error(s) and 4 item(s) reported on remote host
+ End Time:      2010-10-07 21:11:22 (204 seconds)
-----
+ 1 host(s) tested
root@webappsec[nikto-2.1.3]#
```

©2010 C-DAC, Hyderabad

We can scan the target by using the nikto tool with different options. Now let us see how to use the Cgidirs option which is used to scan the target for the Common Gateway Interface directories.

### `./nikto.pl -h <webserver address> -Cgidirs all`

The **-Cgidirs** is the option for Common Gateway Interface directories. CGIDIRS uses hardcoded 404/403 scheme --- it should be dynamic

The **-all** is the option used to scan all CGI directories.

### Findings:

From the figure we can find all the details of the scanned Web Server like IP address (192.168.90.50), Web Server Address (testserver.webappsec.com), port number:80, Web Server name and Version number (Apache /2.2.11) and also we can also find that the Web Server Version is outdated, the Operating system name (Win32), PHP version number (php /5.3.0) which is outdated version, Directory Indexing ...etc.

Once the CGI directories are scanned, the attacker may try to upload a vulnerable CGI script into the server and make his attack successful. Uploading is not very difficult as here we can understand that the target system is based on Microsoft IIS/6.0 and allowed HTTP methods like OPTION, TRACE, GET, HEAD, POST, PUT etc . Using PUT method one can easily upload anything into the server. It also shows that Public HTTP methods, OSVDB etc.





`./nikto.pl -host <webserver address>  
-mutate <options>`

```
Terminal
File Edit View Terminal Tabs Help
root@webappsec[nikto-2.1.3]# ./nikto.pl -h testserver.webappsec.com -mutate 2
- Mutate is deprecated, use -Plugins instead
- Nikto v2.1.3
-----
+ Target IP: 192.168.90.50
+ Target Hostname: testserver.webappsec.com
+ Target Port: 80
+ Using Mutation: Guess for password file names
+ Start Time: 2010-10-07 20:40:00
-----
+ Server: Apache/2.2.11 (Win32) PHP/5.3.0
+ No CGI Directories found (use '-C all' to force check all possible dirs)
+ Apache/2.2.11 appears to be outdated (current is at least Apache/2.2.16). Apache 1.3.42 and 2.0.63 are also current.
+ PHP/5.3.0 appears to be outdated (current is at least 5.3.2)
+ OSVDB-877: HTTP TRACE method is active, suggesting the host is vulnerable to XST
+ OSVDB-3268: /icons/: Directory indexing found.
+ 6417 items checked: 1 error(s) and 4 item(s) reported on remote host
+ End Time: 2010-10-07 20:41:48 (108 seconds)
-----
+ 1 host(s) tested
root@webappsec[nikto-2.1.3]#
```

©2010 C-DAC, Hyderabad

The other option is mutate. The **-mutate** is the option which is used to specify mutation technique. A mutation will cause Nikto to merge tests or attempt to estimate values. These techniques may cause a great amount of tests to be launched against the target machine.

The options are: to test all files with all root directories, to guess for password file names, to enumerate user names via Apache and enumerate user names via cgiwrap.

From the above output, we are able to understand about root directories, target hostname, port and the mutation technique for guess for password file names.

### Example:

**./nikto.pl** is Perl executable Web Server scanning tool

**-h** is the option to scan the specified host

**www.webappsec.com** is the target webserver address

**-mutate** is the option to scan the system using mutate option

**2** is used to guess for password file names

Below are the options which can be used with the **mutate** technique

1 - Test all files with all root directories

2 - Guess for password file names

3 - Enumerate user names via Apache (/~user type requests)

- 4 - Enumerate user names via cgiwrap (/cgi-bin/cgiwrap/~user type requests)
- 5 - Tries to brute force sub-domain names, assume that the host name is the parent domain
- 6 - Tries to guess directory names from the supplied dictionary file



**./nikto.pl -host <webserver address>  
-port <port range>**

```
Terminal
File Edit View Terminal Tabs Help
root@webappsec[nikto-2.1.3]#./nikto.pl -h www.webappsec.com -port 8080-8090
- Nikto v2.1.3
-----
+ No web server found on 192.168.90.100:8081
-----
+ No web server found on 192.168.90.100:8082
-----
+ No web server found on 192.168.90.100:8083
-----
+ No web server found on 192.168.90.100:8084
-----
+ No web server found on 192.168.90.100:8085
-----
+ No web server found on 192.168.90.100:8086
-----
+ No web server found on 192.168.90.100:8087
-----
+ No web server found on 192.168.90.100:8088
-----
+ No web server found on 192.168.90.100:8089
-----
+ No web server found on 192.168.90.100:8090
-----
+ Target IP:      192.168.90.100
```

©2010 C-DAC, Hyderabad

By using nikto tool even we can scan the range of ports or a single port to find out whether any webserver is running on any open port. From the above figure we can observe the scanned ports range from 8080-8090 report of www.webappsec.com server.

#### Example:

**./nikto.pl** is Perl executable Web Server scanning tool

**-h** is the option to scan the specified host

**www.webappsec.com** is the target webserver address

**-port** is the option to scan the range of ports or particular port

**8080-8090** is the range of the ports to be scanned

The output of the above command is explained below

```
root@webappsec[nikto-2.1.3]#./nikto.pl -h www.webappsec.com -port 8080-8090
- Nikto v2.1.3
```

```
-----
+ No web server found on 192.168.90.100:8081
```

```
-----
+ No web server found on 192.168.90.100:8082
```

```
-----
+ No web server found on 192.168.90.100:8083
```

```
-----
+ No web server found on 192.168.90.100:8084
```

```

-----
+ No web server found on 192.168.90.100:8085
-----
+ No web server found on 192.168.90.100:8086
-----
+ No web server found on 192.168.90.100:8087
-----
+ No web server found on 192.168.90.100:8088
-----
+ No web server found on 192.168.90.100:8089
-----
+ No web server found on 192.168.90.100:8090
-----
+ Target IP:      192.168.90.100
+ Target Hostname: www.webappsec.com
+ Target Port:    8080
+ Start Time:     2010-10-07 21:56:16
-----
+ Server: Apache/2.2.9 (Debian) PHP/5.2.6-3 with Suhosin-Patch mod_python/3.3.1
Python/2.5.2 mod_perl/2.0.4 Perl/v5.10.0
+ No CGI Directories found (use '-C all' to force check all possible dirs)
+ Number of sections in the version string differ from those in the database, the server
reports: apache/2.2.9 while the database has: 2.2.16. This may cause false positives.
+ Number of sections in the version string differ from those in the database, the server
reports: php/5.2.6-3 while the database has: 5.3.2. This may cause false positives.
+ PHP/5.2.6-3 appears to be outdated (current is at least 5.3.2)
+ Number of sections in the version string differ from those in the database, the server
reports: python/2.5.2 while the database has: 2.6.10. This may cause false positives.
+ Python/2.5.2 appears to be outdated (current is at least 2.6.10)
+ Number of sections in the version string differ from those in the database, the server
reports: mod_perl/2.0.4 while the database has: 5.8. This may cause false positives.
+ mod_perl/2.0.4 appears to be outdated (current is at least 5.8)
+ Perl/v5.10.0 appears to be outdated (current is at least v5.12.0)
+ Allowed HTTP Methods: GET, HEAD, POST, OPTIONS, TRACE
+ OSVDB-877: HTTP TRACE method is active, suggesting the host is vulnerable to
XST
+ Retrieved x-powered-by header: PHP/5.2.6-3
+ OSVDB-3092: /phpmyadmin/: phpMyAdmin is for managing MySQL databases, and
should be protected or limited to authorized hosts.
+ OSVDB-3092: /manual/: Web server manual found.
+ OSVDB-3268: /icons/: Directory indexing found.
+ OSVDB-3268: /manual/images/: Directory indexing found.
+ OSVDB-3233: /icons/README: Apache default file found.
+ 6417 items checked: 1 error(s) and 16 item(s) reported on remote host
+ End Time:      2010-10-07 21:57:27 (71 seconds)
-----

```

+ 1 host(s) tested

\*\*\*\*\*

Portions of the server's ident string (Apache/2.2.9) are not in  
the Nikto database or is newer than the known string. Would you like  
to submit this information (\*no server specific data\*) to CIRT.net  
for a Nikto update (or you may email to [sullo@cirt.net](mailto:sullo@cirt.net)) (y/n)? n

root@webappsec[nikto-2.1.3]#



`./nikto.pl -host <webserver address>  
-e <options>`

```
Terminal
File Edit View Terminal Tabs Help
root@webappsec[nikto-2.1.3]# ./nikto.pl -h testserver.webappsec.com -e 6
- Nikto v2.1.3
-----
+ Target IP: 192.168.90.50
+ Target Hostname: testserver.webappsec.com
+ Target Port: 80
+ Using IDS Evasion: TAB as request spacer
+ Start Time: 2010-10-07 22:10:44
-----
+ Server: Apache/2.2.11 (Win32) PHP/5.3.0
+ No CGI Directories found (use '-C all' to force check all possible dirs)
+ Apache/2.2.11 appears to be outdated (current is at least Apache/2.2.16). Apache 1.3.42 and 2.0.63 are also current.
+ PHP/5.3.0 appears to be outdated (current is at least 5.3.2)
+ OSVDB-877: HTTP TRACE method is active, suggesting the host is vulnerable to XST
+ OSVDB-3268: /icons/: Directory indexing found.
+ 6417 items checked: 1 error(s) and 4 item(s) reported on remote host
+ End Time: 2010-10-07 22:11:53 (69 seconds)
-----
+ 1 host(s) tested
root@webappsec[nikto-2.1.3]#
root@webappsec[nikto-2.1.3]#
```

©2010 C-DAC, Hyderabad

The **-evasion** techniques is used to specify the LibWhisker IDS evasion technique to use with the following options as Random URI encoding, Directory self-reference, Premature URL ending and Prepend long random string.

#### Example:

**./nikto.pl** is Perl executable Web Server scanning tool

**-h** is the option to scan the specified host

**www.webappsec.com** is the target webserver address

**-e** is the option to scan the target with **evasion** technique

**6** is the option is used to scan the target to TAB as request spacer

Below are the options which can be used with the **evasion** technique

- 1 - Random URI encoding (non-UTF8)
- 2 - Directory self-reference (./.)
- 3 - Premature URL ending
- 4 - Prepend long random string
- 5 - Fake parameter
- 6 - TAB as request spacer
- 7 - Change the case of the URL
- 8 - Use Windows directory separator (\)
- A - Use a carriage return (0x0d) as a request spacer
- B - Use binary value 0x0b as a request spacer



`./nikto.pl -host <webserver address>`  
`-Display <options>`

```
Terminal
File Edit View Terminal Tabs Help
root@webappsec[nikto-2.1.3]# ./nikto.pl -h testserver.webappsec.com -Display 3
- Nikto v2.1.3
+ Target IP: 192.168.90.50
+ Target Hostname: testserver.webappsec.com
+ Target Port: 80
+ Start Time: 2010-10-08 01:52:45
+ Server: Apache/2.2.11 (Win32) PHP/5.3.0
+ No CGI Directories found (use '-C all' to force check all possible dirs)
+ Apache/2.2.11 appears to be outdated (current is at least Apache/2.2.16). Apache 1.3.42 and 2.0.63 are also current.
+ PHP/5.3.0 appears to be outdated (current is at least 5.3.2)
+ OSVDB-877: HTTP TRACE method is active, suggesting the host is vulnerable to XST
+ OSVDB-3268: /icons/: Directory indexing found.
+ /icons/ - 200/OK Response could be Directory indexing found.
+ 6417 items checked: 1 error(s) and 4 item(s) reported on remote host
+ End Time: 2010-10-08 01:53:55 (70 seconds)
+ 1 host(s) tested
root@webappsec[nikto-2.1.3]#
```

©2010 C-DAC, Hyderabad

The **-Display** is the option used to display the output of the nikto scan with following options as show redirects, show cookies received, show all 200/OK responses, show URLs which require authentication, Debug Output and Verbose Output.

### Example:

**./nikto.pl** is Perl executable Web Server scanning tool

**-h** is the option to scan the specified host

**www.webappsec.com** is the target webserver address

**-Display** is the option to scan the target with **Display** technique

**3** is the option is used to scan the target to TAB as request spacer

Below are the options which can be used with the **Display** technique

- 1 - Show redirects
- 2 - Show cookies received
- 3 - Show all 200/OK responses
- 4 - Show URLs which require authentication
- D - Debug Output
- E - Display all HTTP errors
- P - Print progress to STDOUT
- V - Verbose Output



## Nikto.pl -host <IP Address> -Single

```
Terminal
File Edit View Terminal Tabs Help
root@webappsec[nikto-2.1.3]# ./nikto.pl -h testserver.webappsec.com -Single
----- Nikto 2.1.3
----- Single Request Mode
      Hostname or IP: 192.168.90.50
      Port (80): 80
      URI (/): http://testserver.webappsec.com
      SSL (0):
      Proxy host:
      Proxy port:
      Show HTML Response (1):
      HTTP Version (1.1):
      HTTP Method (GET):
      User-Agent (Mozilla/4.75 (Nikto/2.1.3):
      Connection (Keep-Alive):
      Data:
      force_bodysnatch (0):
      force_close (1):
      http_space1 ( ):
      http_space2 ( ):
      include_host_in_uri (0):
      invalid_protocol_return_value (1):
      max_size (0):
      protocol (HTTP):
      require_newline_after_headers (0):
```

©2010 C-DAC, Hyderabad

The **-Single** is the option used to scan for a single host and prompt for all the questions or request details of the target machine like URL proxy host, proxy port etc.

### Example:

```
root@webappsec[nikto-2.1.3]# ./nikto.pl -h testserver.webappsec.com -Single
```

```
----- Nikto 2.1.3
----- Single Request Mode
      Hostname or IP: 192.168.90.50
      Port (80): 80
      URI (/): http://testserver.webappsec.com
      SSL (0):
      Proxy host:
      Proxy port:
      Show HTML Response (1):
      HTTP Version (1.1):
      HTTP Method (GET):
      User-Agent (Mozilla/4.75 (Nikto/2.1.3):
      Connection (Keep-Alive):
      Data:
      force_bodysnatch (0):
      force_close (1):
      http_space1 ( ):
```



http\_space2 ( ):  
include\_host\_in\_uri (0):  
invalid\_protocol\_return\_value (1):  
max\_size (0):  
protocol (HTTP):  
require\_newline\_after\_headers (0):  
retry (0):  
ssl\_save\_info (0):  
timeout (10):  
uri\_password ():  
uri\_postfix ():  
uri\_prefix ():  
uri\_user ():  
Enable Anti-IDS (0):

----- **Done with questions**

Host Name: testserver.webappsec.com  
Host IP: 192.168.90.50  
HTTP Response Code: 403

----- **Connection Details**

Connection: Keep-Alive  
Host: 192.168.90.50  
User-Agent: Mozilla/4.75 (Nikto/2.1.3  
data:  
force\_bodysnatch: 0  
force\_close: 1  
force\_open: 0  
host: 192.168.90.50  
http\_space1:  
http\_space2:  
ignore\_duplicate\_headers: 1  
include\_host\_in\_uri: 0  
invalid\_protocol\_return\_value: 1  
max\_size: 0  
method: GET  
port: 80  
protocol: HTTP  
require\_newline\_after\_headers: 0  
retry: 0  
ssl: 0  
ssl\_save\_info: 0  
timeout: 10  
trailing\_slurp: 0  
uri: http://testserver.webappsec.com  
uri\_param\_sep: ?  
uri\_postfix:  
uri\_prefix:

version: 1.1

----- **Response Headers**

Connection: Keep-Alive  
Content-Length: 202  
Content-Type: text/html; charset=iso-8859-1  
Date: Wed, 06 Oct 2010 10:09:43 GMT  
Keep-Alive: timeout=5, max=100  
Server: Apache/2.2.11 (Win32) PHP/5.3.0  
code: 403  
http\_data\_sent: 1  
http\_eol:  
http\_space1:  
http\_space2:  
message: Forbidden  
protocol: HTTP  
uri: http://testserver.webappsec.com  
version: 1.1

----- **Response Content**

```
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>403 Forbidden</title>
</head><body>
<h1>Forbidden</h1>
<p>You don't have permission to access /
on this server.</p>
</body></html>
root@webappsec[nikto-2.1.3]#
```



## Nikto Other Options

- `./nikto.pl -h <IP Address> - root`
- `./nikto.pl -h <IP Address> -ssl`
- `./nikto.pl -h <IP Address> -dbcheck`
- `./niktp.pl -h<IP Address> -vhost`

©2010 C-DAC, Hyderabad

The **-root** used for the beginning of every request.

This is helpful to test applications or web servers which have all of their files under a certain directory.

The **-ssl** is used for to test SSL on the ports specified. Using this option will speed up requests to HTTPS ports, since otherwise the HTTP request will have to timeout first.

The **-dbcheck** to check the scan databases for syntax errors.

The **-vhost** specify the Host header to be sent to the target.



# Nessus

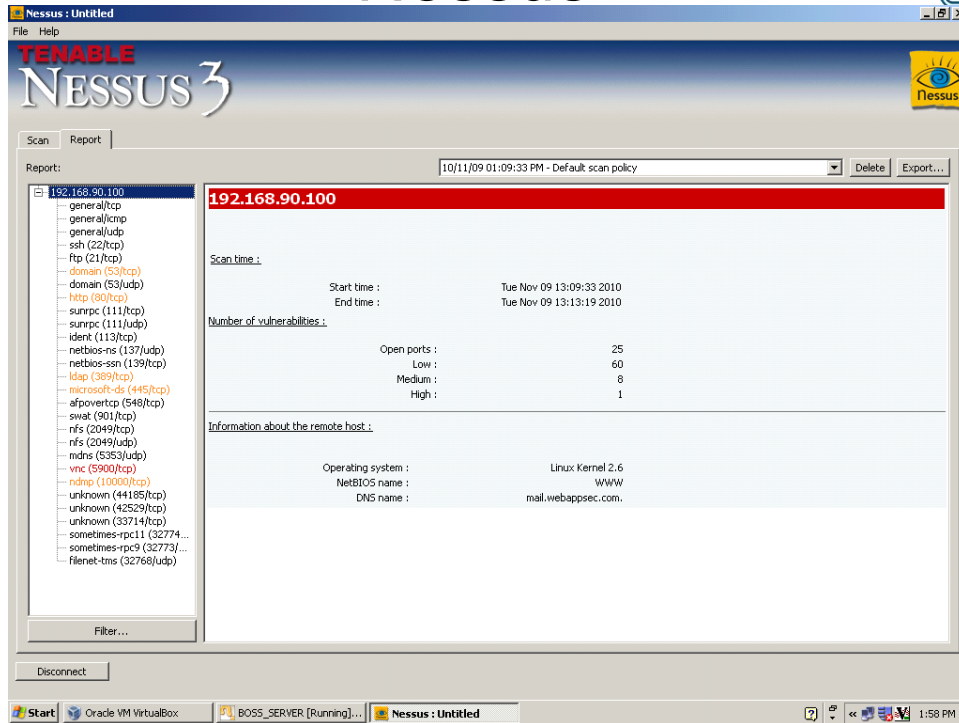
- Tenable Network Scanner
- Automated Vulnerability scanner
- Detects potential vulnerabilities in the web Server
- Checks for
  - » Misconfigurations, default passwords, blank passwords
- Performs
  - Dictionary attack, Denial of Service attack

©2010 C-DAC, Hyderabad

Nessus is an open source tool. Its goal is to detect and analyze the vulnerabilities of the Web Server. It check for the vulnerabilities that allow a remote hacker or cracker to control or access sensitive data on a system, and checks for misconfiguration (example missing patches, e-Mail configurations etc). It also tests the Server whether it is accepting the Default passwords, blank passwords, common passwords. It is also having the capability of launching a dictionary attack, Denials of service against the TCP/IP stack by using jumbled packets.



# Nessus



©2010 C-DAC, Hyderabad

## General/TCP

**Pingtheremotehost** The remote host is up

NessusID: [10180](#) TCP timestamps

**Synopsis** :

The remote service implements TCP timestamps.

**Description** :

The remote host implements TCP timestamps, as defined by RFC1323. A side effect of this feature is that the uptime of the remote host can sometimes be computed.

See also:

<http://www.ietf.org/rfc/rfc1323.txt>

**Risk factor** :

None

Nessus ID : [25220](#) Host FQDN 192.168.90.100 resolves as mail.webappsec.com.

Nessus ID	:	<a href="#">12053</a>	OS	Identification
Remote operating system	:	Linux	Kernel	2.6
Confidence Level	:			70
Method	:			SinFP

The remote host is running Linux Kernel 2.6

Nessus ID : [11936](#) **Information about the scan** Information about this scan :

Nessus version	:	3.2.1.1
Plugin feed version	:	\$Date: 2005/11/08 13:18:41 \$
Type of plugin feed	:	CVS
Scanner IP	:	192.168.90.50
Port scanner(s)	:	synscan
Port range	:	default
Thorough tests	:	no
Experimental tests	:	no
Paranoia level	:	1
Report Verbosity	:	1
Safe checks	:	yes
Optimize the test	:	yes
Max hosts	:	20
Max checks	:	5
Recv timeout	:	5
Scan Start Date	:	2010/11/9 13:09
Scan duration	:	222 sec

Nessus ID : [19506](#)

### General/ICMP

<b>icmp</b>	<b>timestamp</b>	<b>request</b>
<b>Synopsis</b>		:

It is possible to determine the exact time set on the remote host.

**Description** :

The remote host answers to an ICMP timestamp request. This allows an attacker to know the date which is set on your machine.

This may help him to defeat all your time based authentication protocols.

**Solution** :

Filter out the ICMP timestamp requests (13), and the outgoing ICMP timestamp replies (14).

**Risk factor** :

None

**Plugin output** :

The difference between the local and remote clocks is -19799 seconds

CVE : CVE-1999-0524

Nessus ID : [10114](#)

### **General/UDP**

**Traceroute**For your information, here is the traceroute from 192.168.90.50 to 192.168.90.100 :

192.168.90.50  
192.168.90.100

Nessus ID : [10287](#)

### **SSH (22/TCP)**

**Service detection**An SSH server is running on this port.

Nessus ID : [22964](#) **SSH Server type and version** :  
**Synopsis** :

An SSH server is listening on this port.

**Description** :

It is possible to obtain information about the remote SSH server by sending an empty authentication request.

**Risk factor** :

None

**Plugin output** :

SSH version : SSH-2.0-OpenSSH\_5.1p1 Debian-2  
SSH supported authentication : publickey,password



# nmap

- Network Mapper
- Free Network Scanner
- Determines the details of the target system
  - » Operating System
  - » Version
  - » Open Ports
  - » and also Identifies potential Vulnerabilities

©2010 C-DAC, Hyderabad

**Nmap** uses raw IP packets to find or determine what hosts are available on the network, services (application name and version) those hosts are offering, operating systems (and OS versions) those hosts are running, type of packet filters/firewalls that are in use, and dozens of other characteristics.

## **Information regarding nmap tool :**

Nmap can be downloaded from site <http://nmap.org> and the usage of it is

Usage: nmap [Scan Type(s)] [Options] {target specification}

### **TARGET SPECIFICATION:**

### **HOST DISCOVERY:**

### **SCAN TECHNIQUES:**

### **PORT SPECIFICATION AND SCAN ORDER:**

### **SERVICE/VERSION DETECTION:**

### **SCRIPT SCAN:**

### **OS DETECTION:**

### **TIMING AND PERFORMANCE:**

### **FIREWALL/IDS EVASION AND SPOOFING:**

### **OUTPUT:**

### **MISC:**

### **EXAMPLES:**

```
nmap -v -A scanme.nmap.org
```

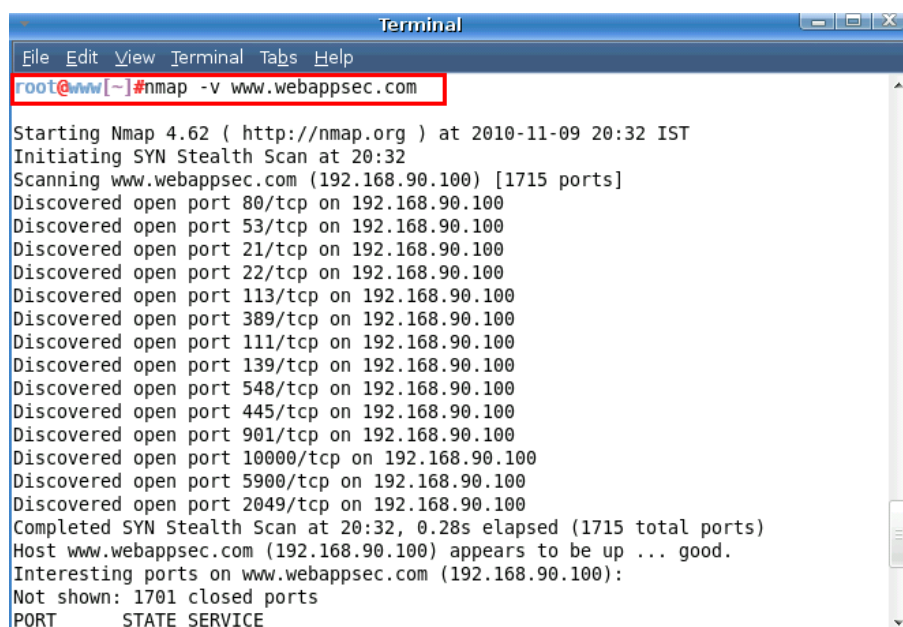
```
nmap -v -sP 192.168.0.0/16 10.0.0.0/8
```

```
nmap -v -iR 10000 -PN -p 80
```





## Nmap port scan in Verbose mode



```
Terminal
File Edit View Terminal Tabs Help
root@www[-]#nmap -v www.webappsec.com

Starting Nmap 4.62 ( http://nmap.org ) at 2010-11-09 20:32 IST
Initiating SYN Stealth Scan at 20:32
Scanning www.webappsec.com (192.168.90.100) [1715 ports]
Discovered open port 80/tcp on 192.168.90.100
Discovered open port 53/tcp on 192.168.90.100
Discovered open port 21/tcp on 192.168.90.100
Discovered open port 22/tcp on 192.168.90.100
Discovered open port 113/tcp on 192.168.90.100
Discovered open port 389/tcp on 192.168.90.100
Discovered open port 111/tcp on 192.168.90.100
Discovered open port 139/tcp on 192.168.90.100
Discovered open port 548/tcp on 192.168.90.100
Discovered open port 445/tcp on 192.168.90.100
Discovered open port 901/tcp on 192.168.90.100
Discovered open port 10000/tcp on 192.168.90.100
Discovered open port 5900/tcp on 192.168.90.100
Discovered open port 2049/tcp on 192.168.90.100
Completed SYN Stealth Scan at 20:32, 0.28s elapsed (1715 total ports)
Host www.webappsec.com (192.168.90.100) appears to be up ... good.
Interesting ports on www.webappsec.com (192.168.90.100):
Not shown: 1701 closed ports
PORT      STATE SERVICE
```

©2010 C-DAC, Hyderabad

By using nmap tool one can find the port details along with the service name in the verbose mode of the target Web Server by using the option `-v`. We can observe the output of the scanned web server `www.webappsec.com` in the slide and the complete output is mentioned below.

### Findings:

#### # `nmap -v www.webappsec.com`

Starting Nmap 4.62 ( http://nmap.org ) at 2010-11-09 20:36 IST

Initiating SYN Stealth Scan at 20:36

Scanning www.webappsec.com (192.168.90.100) [1715 ports]

Discovered open port 80/tcp on 192.168.90.100

Discovered open port 22/tcp on 192.168.90.100

Discovered open port 113/tcp on 192.168.90.100

Discovered open port 53/tcp on 192.168.90.100

Discovered open port 389/tcp on 192.168.90.100

Discovered open port 21/tcp on 192.168.90.100

Discovered open port 2049/tcp on 192.168.90.100

Discovered open port 445/tcp on 192.168.90.100

Discovered open port 139/tcp on 192.168.90.100

Discovered open port 10000/tcp on 192.168.90.100

Discovered open port 548/tcp on 192.168.90.100

Discovered open port 111/tcp on 192.168.90.100

Discovered open port 901/tcp on 192.168.90.100  
Discovered open port 5900/tcp on 192.168.90.100  
Completed SYN Stealth Scan at 20:36, 0.23s elapsed (1715 total ports)  
Host www.webappsec.com (192.168.90.100) appears to be up ... good.  
Interesting ports on www.webappsec.com (192.168.90.100):  
Not shown: 1701 closed ports  
PORT STATE SERVICE  
21/tcp open ftp  
22/tcp open ssh  
53/tcp open domain  
80/tcp open http  
111/tcp open rpcbind  
113/tcp open auth  
139/tcp open netbios-ssn  
389/tcp open ldap  
445/tcp open microsoft-ds  
548/tcp open afp  
901/tcp open samba-swat  
2049/tcp open nfs  
5900/tcp open vnc  
10000/tcp open snet-sensor-mgmt  
  
Read data files from: /usr/share/nmap  
Nmap done: 1 IP address (1 host up) scanned in 0.291 seconds  
Raw packets sent: 1715 (75.460KB) | Rcvd: 3444 (144.676KB)



# Service Version Identification

```
Terminal
File Edit View Terminal Tabs Help
root@www[~]# nmap -sV www.webappsec.com

Starting Nmap 4.62 ( http://nmap.org ) at 2010-11-09 21:29 IST
Interesting ports on www.webappsec.com (192.168.90.100):
Not shown: 1701 closed ports
PORT      STATE SERVICE      VERSION
21/tcp    open  ftp          ProFTPD 1.3.1
22/tcp    open  ssh          (protocol 2.0)
53/tcp    open  domain
80/tcp    open  http         Apache httpd 2.2.9 ((Debian) PHP/5.2.6-3 with Suhosin-Patch mod_python/3.3.1 Python/2.5.2 mod_perl/2.0.4 Perl/v5.10.0)
111/tcp   open  rpcbind
113/tcp   open  ident
139/tcp   open  netbios-ssn Samba smbd 3.X (workgroup: WORKGROUP)
389/tcp   open  ldap         OpenLDAP 2.2.X
445/tcp   open  netbios-ssn Samba smbd 3.X (workgroup: WORKGROUP)
548/tcp   open  afp?
901/tcp   open  http         Samba SWAT administration server
2049/tcp  open  rpcbind
5900/tcp  open  vnc          VNC (protocol 3.7)
10000/tcp open  http         Webmin httpd
1 service unrecognized despite returning data. If you know the service/version,
please submit the following fingerprint at http://www.insecure.org/cgi-bin/servicefp-submit.cgi :

```

©2010 C-DAC, Hyderabad

By using advanced option - sV in nmap tool one can find the port details along with the service name and version of the service running on the target Web. We can observe the output of the scanned web server www.webappsec.com in the slide and the complete output is mentioned below.

## Findings:

# nmap -sV www.webappsec.com

Starting Nmap 4.62 ( http://nmap.org ) at 2010-11-09 21:33 IST

Interesting ports on www.webappsec.com (192.168.90.100):

Not shown: 1701 closed ports

PORT	STATE	SERVICE	VERSION
------	-------	---------	---------

21/tcp	open	ftp	ProFTPD 1.3.1
--------	------	-----	---------------

22/tcp	open	ssh	(protocol 2.0)
--------	------	-----	----------------

53/tcp	open	domain	
--------	------	--------	--

80/tcp	open	http	Apache httpd 2.2.9 ((Debian) PHP/5.2.6-3 with Suhosin-Patch mod_python/3.3.1 Python/2.5.2 mod_perl/2.0.4 Perl/v5.10.0)
--------	------	------	--

111/tcp	open	rpcbind	
---------	------	---------	--

113/tcp	open	ident	
---------	------	-------	--

139/tcp	open	netbios-ssn	Samba smbd 3.X (workgroup: WORKGROUP)
---------	------	-------------	---------------------------------------

389/tcp	open	ldap	OpenLDAP 2.2.X
---------	------	------	----------------

445/tcp	open	netbios-ssn	Samba smbd 3.X (workgroup: WORKGROUP)
---------	------	-------------	---------------------------------------

548/tcp open afp?  
901/tcp open http Samba SWAT administration server  
2049/tcp open rpcbind  
5900/tcp open vnc VNC (protocol 3.7)  
10000/tcp open http Webmin httpd  
1 service unrecognized despite returning data. If you know the service/version, please  
submit the following fingerprint at <http://www.insecure.org/cgi-bin/servicefp-submit.cgi> :  
SF-Port22-TCP:V=4.62%I=7%D=11/9%Time=4CD970E1%P=i686-pc-linux-gnu  
%r(NULL,2  
SF:0,"SSH-2\0-OpenSSH\_5\0.1p1\0Debian-2\r\n");  
Service Info: OS: Unix

Host script results:

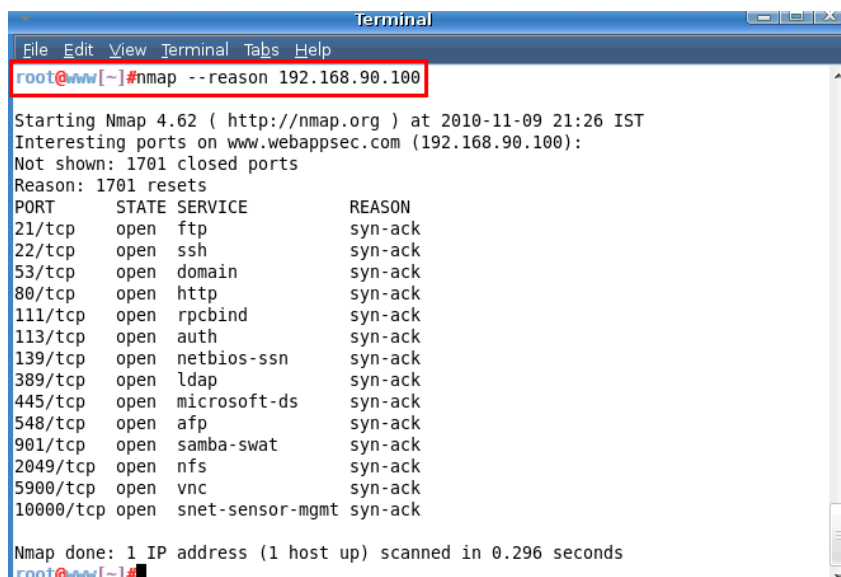
\_ Discover OS Version over NetBIOS and SMB: Unix

Service detection performed. Please report any incorrect results at  
<http://nmap.org/submit/>.

Nmap done: 1 IP address (1 host up) scanned in 16.864 seconds



# Host & port state reasons



```
Terminal
File Edit View Terminal Tabs Help
root@www[~]# nmap --reason 192.168.90.100

Starting Nmap 4.62 ( http://nmap.org ) at 2010-11-09 21:26 IST
Interesting ports on www.webappsec.com (192.168.90.100):
Not shown: 1701 closed ports
Reason: 1701 resets
PORT      STATE SERVICE      REASON
21/tcp    open  ftp          syn-ack
22/tcp    open  ssh          syn-ack
53/tcp    open  domain       syn-ack
80/tcp    open  http         syn-ack
111/tcp   open  rpcbind      syn-ack
113/tcp   open  auth         syn-ack
139/tcp   open  netbios-ssn  syn-ack
389/tcp   open  ldap         syn-ack
445/tcp   open  microsoft-ds syn-ack
548/tcp   open  afp          syn-ack
901/tcp   open  samba-swat   syn-ack
2049/tcp  open  nfs          syn-ack
5900/tcp  open  vnc          syn-ack
10000/tcp open  snet-sensor-mgmt syn-ack

Nmap done: 1 IP address (1 host up) scanned in 0.296 seconds
root@www[~]#
```

©2010 C-DAC, Hyderabad

## --reason (Host and port state reasons)

--reason option is used to show the state of each port which is set to a specific state and the reason each host is up or down. It will display the type of the packet that determined a port or hosts state. This can be observed from the above figure.



# BurpSuite

- Platform for Web Application Security Testing
- Gives full control
- Contains various tools to test the application vulnerability

©2010 C-DAC, Hyderabad

Burp Suite is a best platform used for attacking the web applications, which contains a number of variety tools with many interfaces. All the tools use the same framework for handling and displaying the HTTP messages, authentication, proxies, logging, alerting and extensibility.

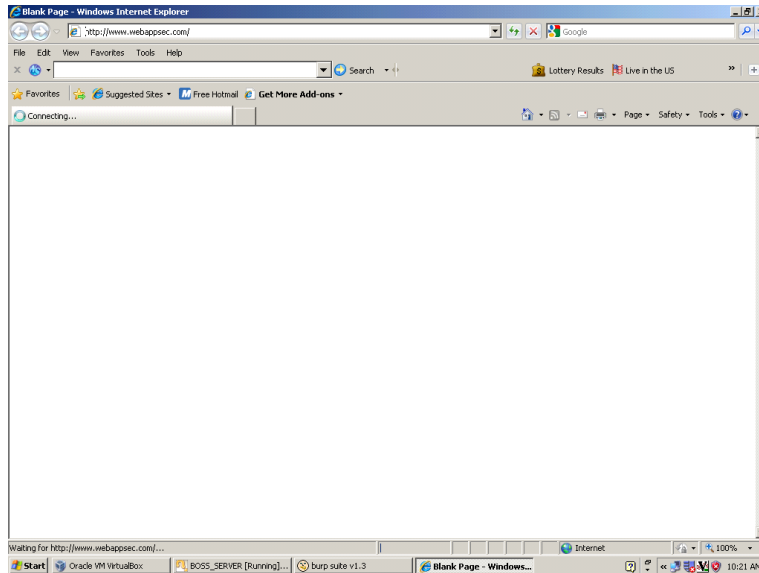
The framework contains the following tools.

- Proxy : Is used as an capturing or intercepting HTTP or HTTPS proxy server which functions as a man-in-the-middle between the end browser and the target web application, by allowing you to intercept, inspect and modify the raw traffic passing in both the directions.
- Spider: Is used for enumerating the entire application's content.
- Scanner: Is used for discovery of vulnerabilities in the application
- Intruder: Is used for web application attacks, for gathering common vulnerabilities ..etc
- Repeater: Is used for manually manipulating and re-issuing the individual HTTP requests, and analyzing the application's responses
- Sequencer: Is used for analyzing the quality of randomness in an application's session tokens
- Decoder: Is used for manual or intelligent decoding and encoding of application data
- Comparer: Is used for performing a visual "diff" between any two items of data

For more details about burp suite visit the link <http://portswigger.net/burp/help/>



# Example: BurpSuite

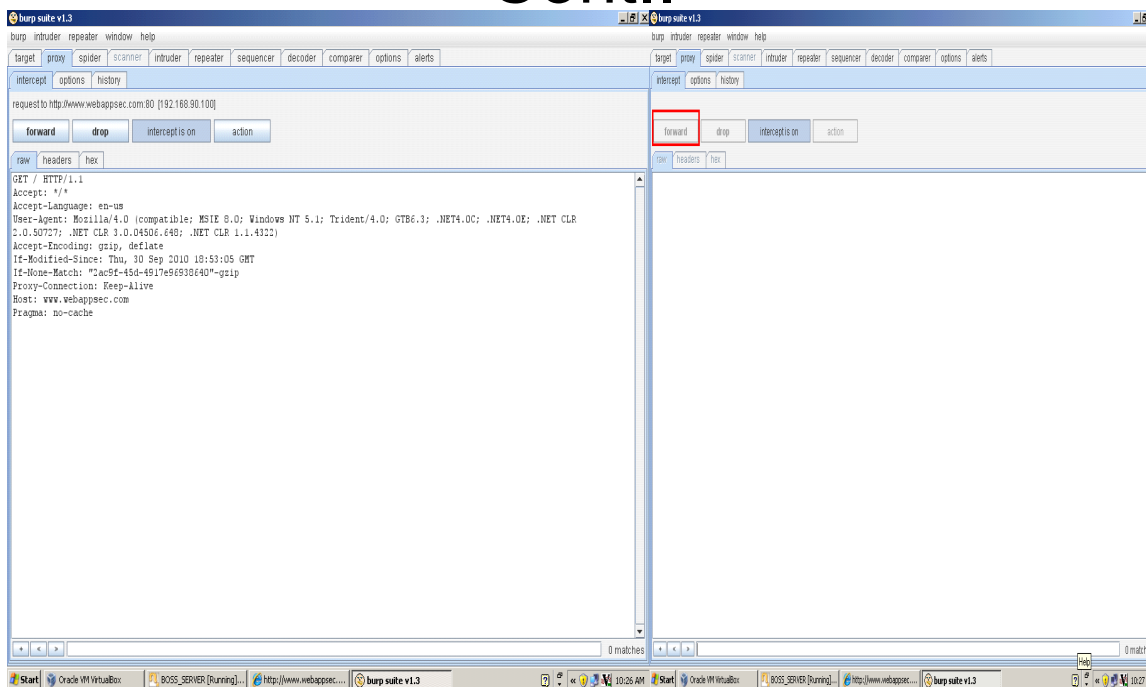


©2010 C-DAC, Hyderabad

From the above figure we can observe that the user is trying to access the website application [www.webappsec.com](http://www.webappsec.com), but he/she is unable to get the front page. The reason behind is that the attacker who is using burpsuite is capturing the data in the middle, if the person (attacker) allows the access to the user then he will be able to view the front page of the site [www.webappsec.com](http://www.webappsec.com). In the next screenshot you can observe how the attacker is able to capture the application data by using burpsuite.



# Cont..



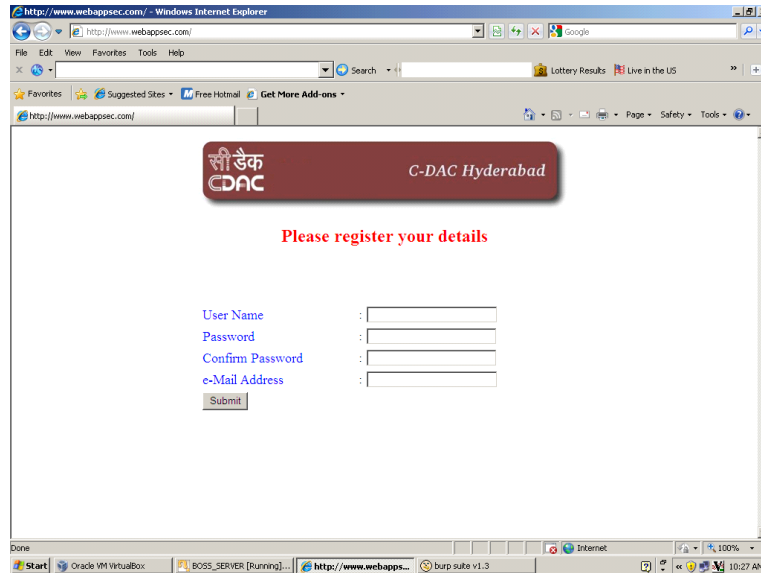
©2010 C-DAC, Hyderabad

From the above figure we can observe that the attacker has captured all the html data by using burpsuite proxy setup and intercept button. If the attacker clicks on the forward button then the details will be submitted to the real server and the response will be displayed to the user who is accessing the [www.webappsec.com](http://www.webappsec.com). In the next slide you can find the screenshot of the front page of the **www.webappsec.com**.





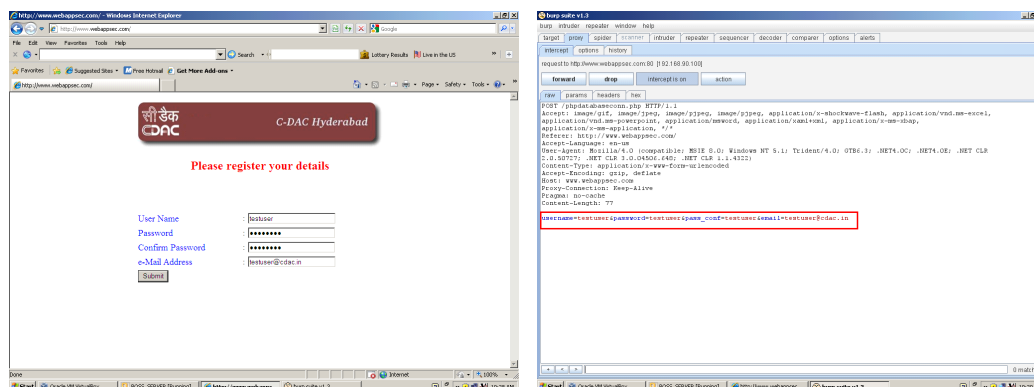
## Cont...



©2010 C-DAC, Hyderabad

From the above figure we can observe that the front page is displayed to the user who has access the [www.webappsec.com](http://www.webappsec.com). Now whatever the user enters into the form it goes to the server through the burpsuite proxy and the attacker who is sitting in the middle will be able to see all the data like user credentials in the clear text form. This can be observed in the next slide screenshot.

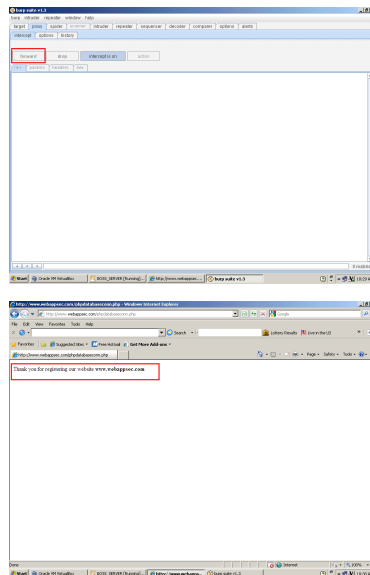
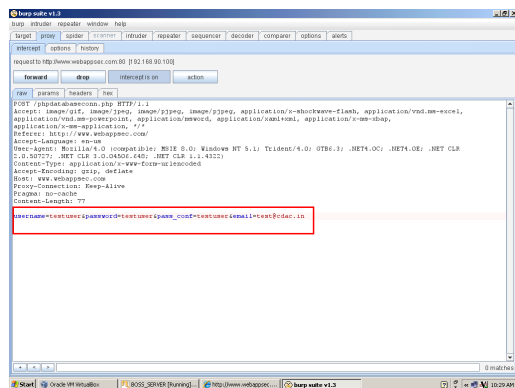
# Cont...



©2010 C-DAC, Hyderabad

From the second figure (highlighted area) we can find that the attacker is able to view all the data which is entered in the form by the user. The details won't be submitted to the server until and unless the attacker clicks on the forward button, because as the details are traveling through burpsuite proxy. The attacker can directly submit the details to the server or he/she can modify them and submit it by clicking on the forward button. This can be observed from the next figure which is in the next slide.

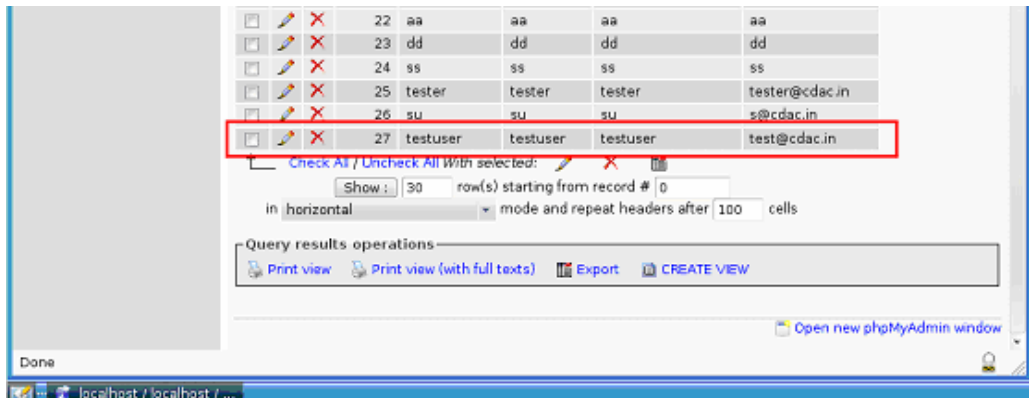
# Cont..



©2010 C-DAC, Hyderabad

From the first figure we can find that the attacker has changed the credentials and has submitted the form to server by clicking on the forward button. From the 3rd screenshot we can observe that the response message from the server. If we check the database in the server we can find that the details which was entered by the attacker will be stored, this can be observed from the next figure in the next slide.

## Cont...



22	aa	aa	aa	aa
23	dd	dd	dd	dd
24	ss	ss	ss	ss
25	tester	tester	tester	tester@cdac.in
26	su	su	su	s@cdac.in
27	testuser	testuser	testuser	test@cdac.in

Check All / Uncheck All With selected: ☐ ☒ ☐

Show: 30 row(s) starting from record # 0

in horizontal mode and repeat headers after 100 cells

Query results operations

[Print view](#) [Print view \(with full texts\)](#) [Export](#) [CREATE VIEW](#)

[Open new phpMyAdmin window](#)

Done

localhost / localhost / ...

©2010 C-DAC, Hyderabad

From the above figure we can observe that the details which were entered by the attackers are stored in the server database. In this way the attacker can hack your Web application.

# Conclusion

- In this module we have discussed about the topics
  - » What is Vulnerability?
  - » Different types of Vulnerabilities?
  - » Vulnerability Discovery
  - » Vulnerability Management
  - » Vulnerability Scanning tools

©2010 C-DAC, Hyderabad

In this module we has discussed about the topics what is Vulnerability, what are the different types of Vulnerabilities, what is Vulnerability Discovery and Management, how to scan the Web Application and find the Vulnerabilities using scanning tools like GooLag, Wikto, Nikto, Nessus, Nmap, Burpsuite...etc.

In the next module we will discuss about Cross Site Scripting and Cross Site Request Forgery.