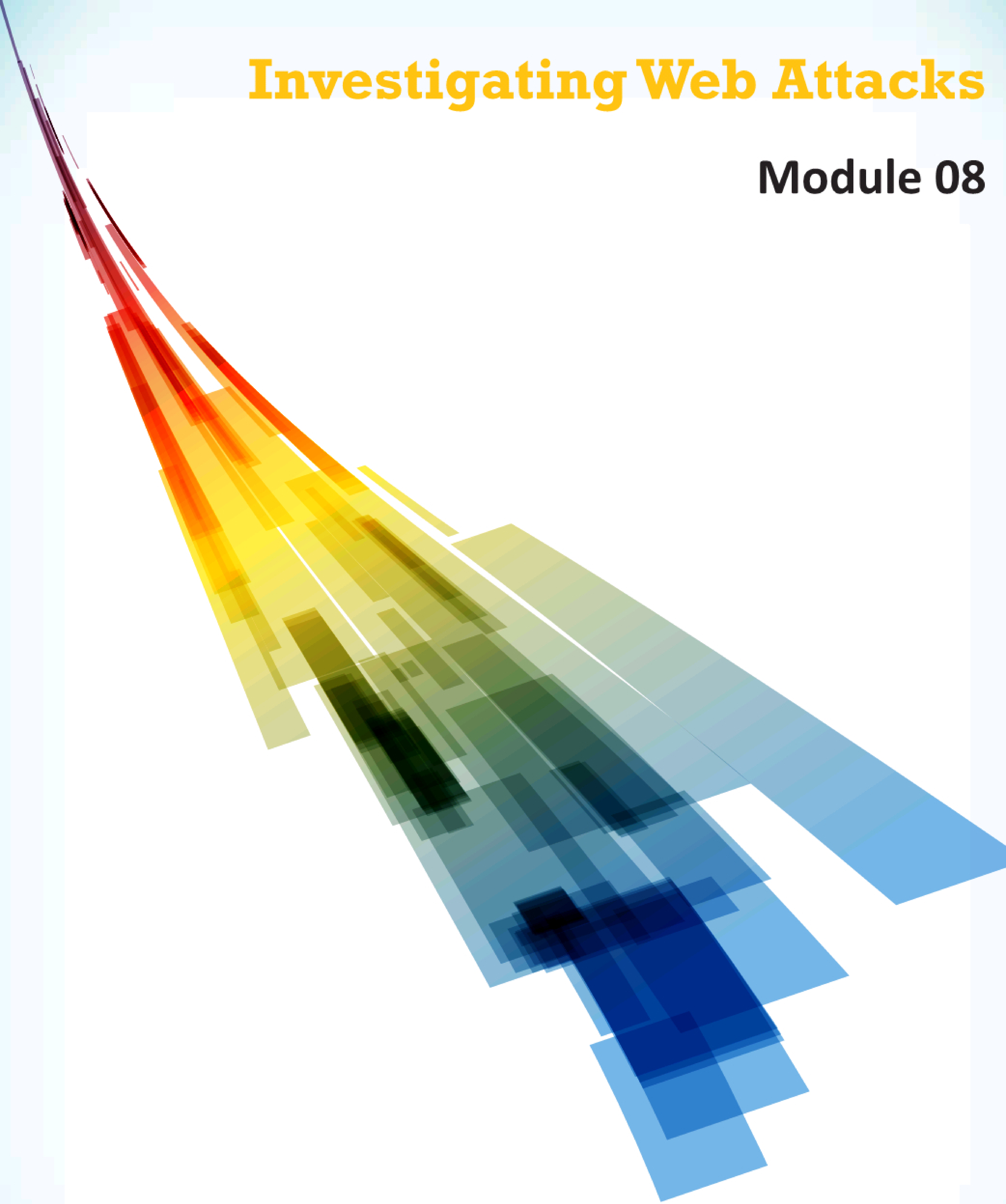
Investigating Web Attacks

Module 08



Investigating Web Attacks

Module 08

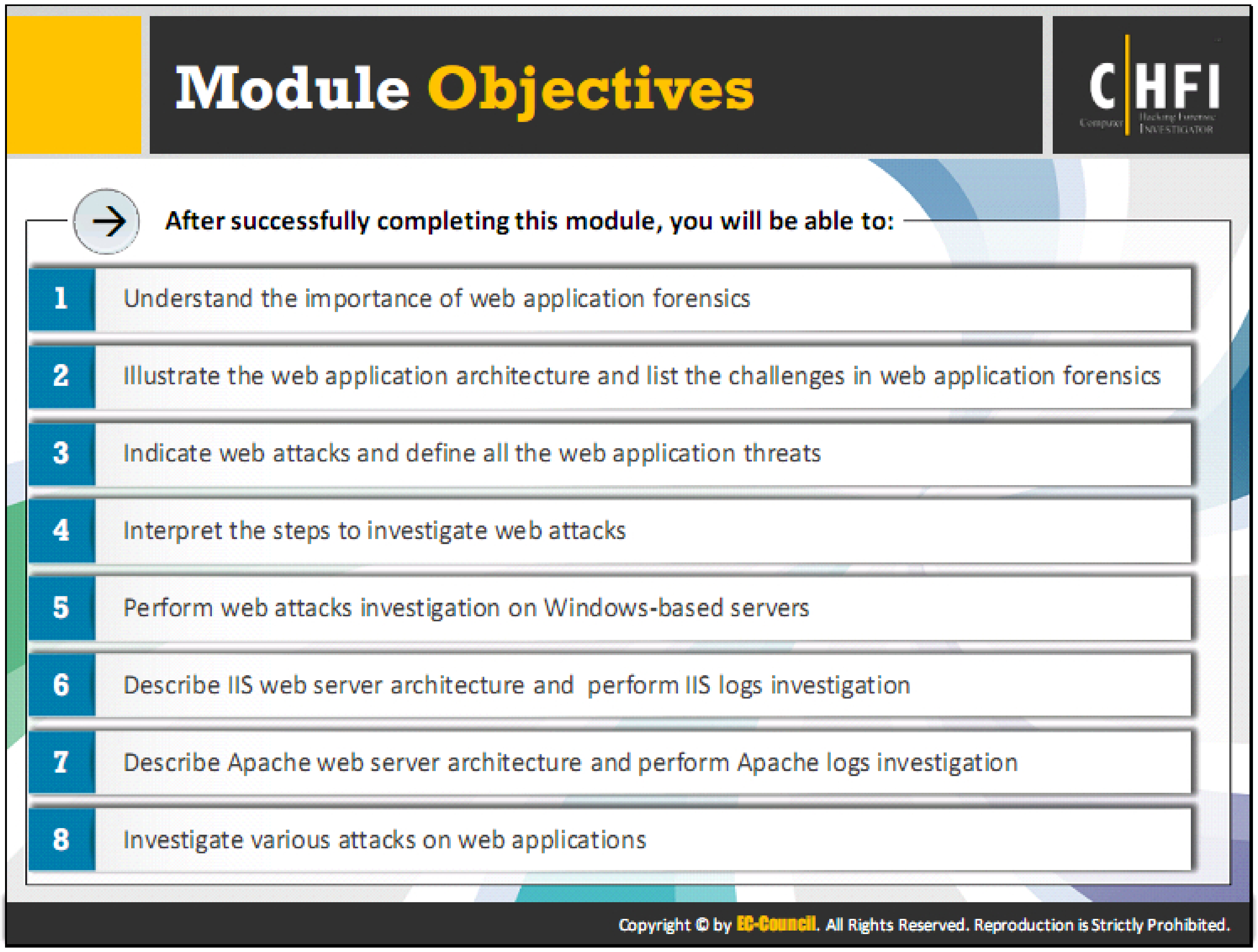
Designed by Cyber Crime Investigators. Presented by Professionals.

CHFI
Computer Hacking Forensic Investigator
1010101

Computer Hacking Forensic Investigator v9

Module 08: Investigating Web Attacks

Exam 312-49

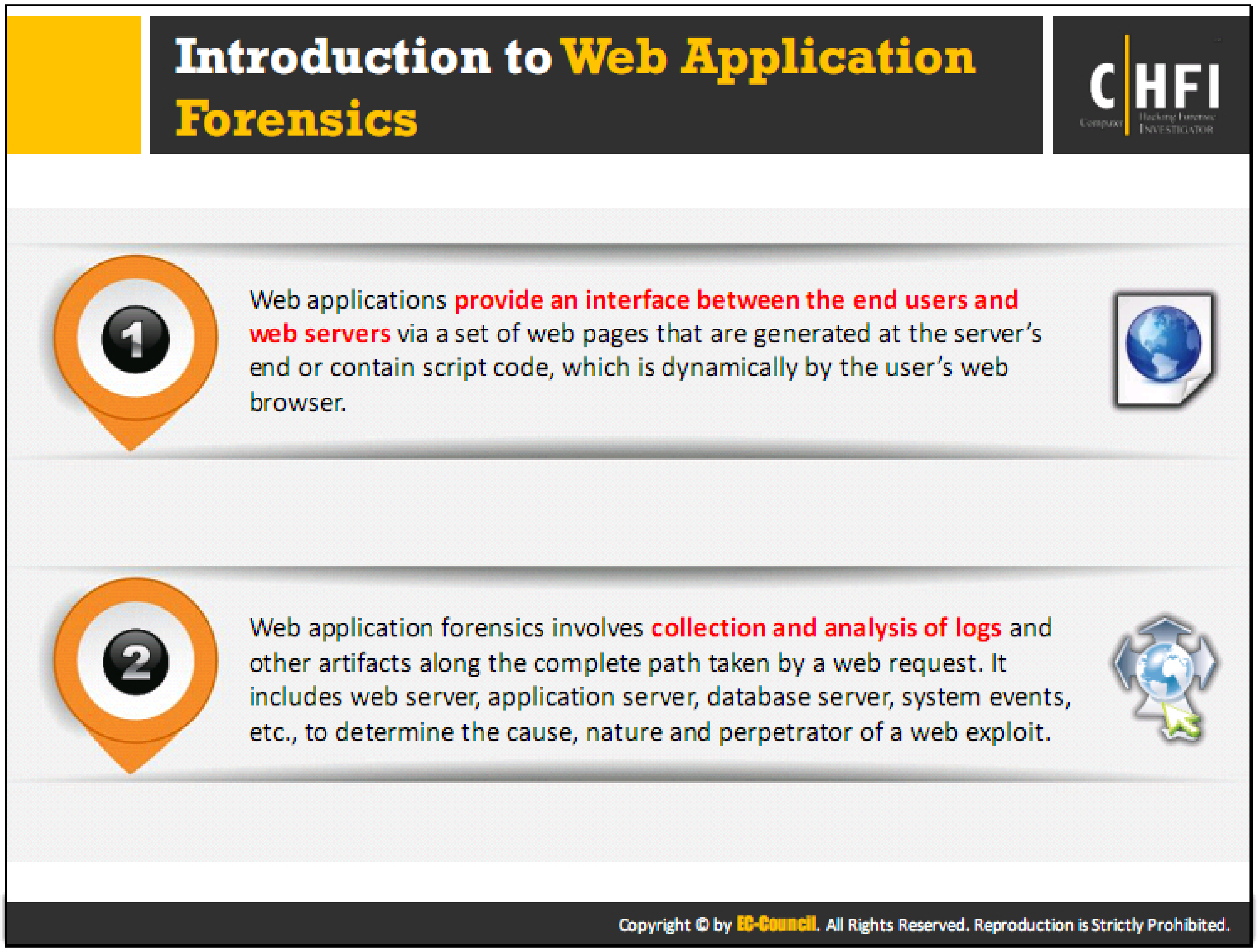


Module Objectives

After successfully completing this module, you will be able to:

1. Understand the importance of web application forensics
2. Illustrate the web application architecture and list the challenges in web application forensics
3. Indicate web attacks and define all the web application threats
4. Interpret the steps to investigate web attacks
5. Perform web attacks investigation on Windows-based servers
6. Describe IIS web server architecture and perform IIS logs investigation
7. Describe Apache web server architecture and perform Apache logs investigation
8. Investigate various attacks on web applications

Web applications allow users to access their resources through client side applications such as web browsers. Some of these web applications may contain vulnerabilities, which can allow attackers to perform attacks, such as SQL Injection, Cross Site Scripting, Local File Inclusion (LFI), Remote File Inclusion (RFI), etc., which leads to either partial or complete damage of the underlying servers. This module discusses numerous types of attacks on web servers and applications. Also, it explains the usage of different tools to identify and investigate such web attacks. This module will familiarize you with:



Introduction to Web Application Forensics

1. Web applications **provide an interface between the end users and web servers** via a set of web pages that are generated at the server's end or contain script code, which is dynamically by the user's web browser.

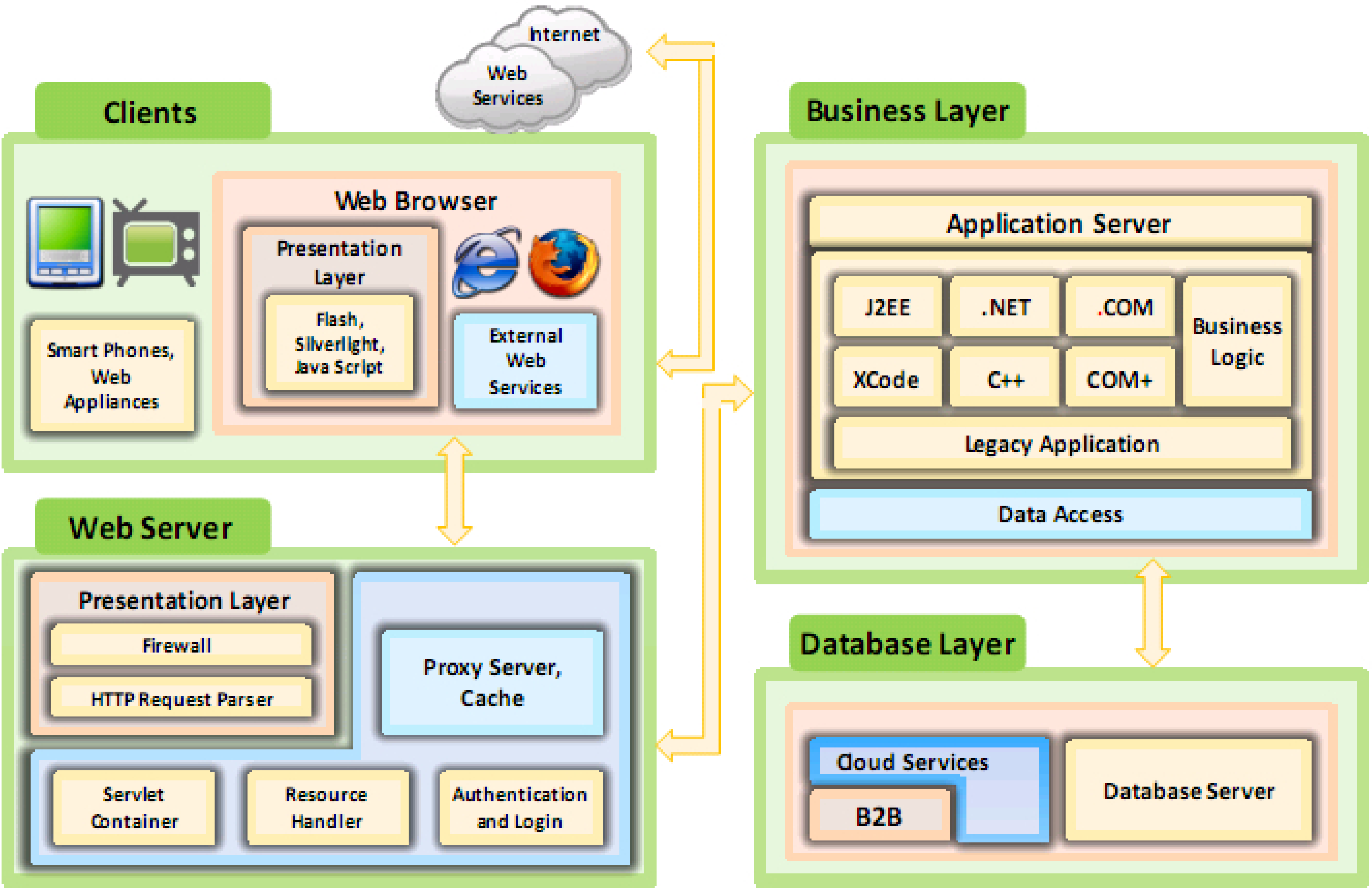
2. Web application forensics involves **collection and analysis of logs** and other artifacts along the complete path taken by a web request. It includes web server, application server, database server, system events, etc., to determine the cause, nature and perpetrator of a web exploit.

Web applications are programs that exist on a central server permitting a user, who visits a website via the Internet, to submit and retrieve data to and from a database. A web application makes a request through a web server. When the server responds to the request, the web application generates documents of the response for better client/user service. The web documents generated by web applications are in a standard format, i.e. HTML, XML, etc., which is supported by all types of browsers. Web applications accomplish the requested task irrespective of the operating system and browsers installed.

Despite having the advantage that the web applications possess, they tend to fall prey for attackers due to improper coding or security monitoring. The attackers try to exploit the vulnerabilities in the coding and gain access to the database contents, thereby gaining sensitive information, such as user credentials, bank account details, etc. Some of the attacks performed on the web applications include SQL injection, cross-site scripting, session hijacking, local and remote file inclusions, remote code execution, etc.

Web application forensics comes into picture when such kinds of attacks occur on web applications. The web application forensics involves forensic examination of web applications and its contents (such as logs, www directory, and config files) to trace back the attack, identify the origin of the attack, and determine how the attack was propagated along with the devices used (mobiles and computer) and the persons involved to perform the attack. The investigators examine the logs and configuration files associated with web server and application server, server side scripts used by the web application, and logs pertaining to third party software applications and operating system, to get an insight of the attack.

Web Application Architecture



Internet
Web Services
Clients
Smart Phones, Web Appliances
Web Browser
Presentation Layer
Flash, Silverlight, Java Script
External Web Services
Web Server
Presentation Layer
Firewall
HTTP Request Parser
Proxy Server, Cache
Servlet Container
Resource Handler
Authentication and Login
Business Layer
Application Server
J2EE
.NET
.COM
XCode
C++
COM+
Business Logic
Legacy Application
Data Access
Database Layer
Cloud Services
B2B
Database Server

All web applications are executed via a support client, i.e. a web browser. Web applications use a group of client-side scripts, such as HTML, JavaScript, etc., which presents the information, and the server-side scripts, such as ASP, PHP, etc., which handles the hardware tasks such as storing and gathering of the required data, are used by the web application for its execution.

In the web application architecture mentioned above, the clients use different web browsers, devices, and external web services with the Internet for execution of the application through different scripting languages. The data access is handled by the database layer using cloud server and the database server. It is important to note that the web server, application server, and database server may either run on independent servers/machines or the same one.

The web application architecture comprises of four layers:

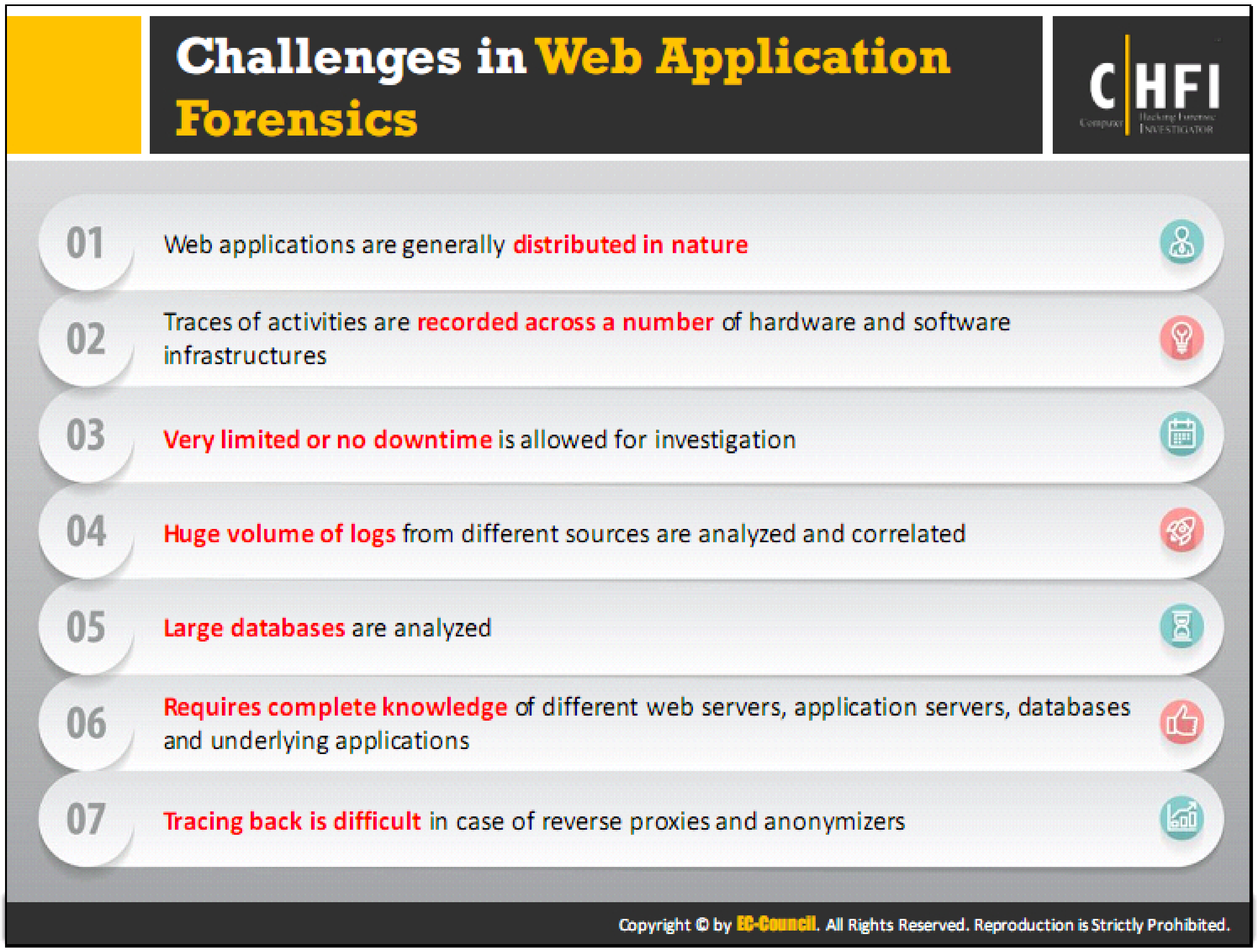
- Clients or Users Layer
- Web Server Layer
- Business Layer
- Database Layer

The client layer includes all the web appliances, such as smartphones and PCs, using which a user interacts with a web application deployed on a web server. The user requests for a website by entering a URL in the web browser and the request traverses to the web server. The web server responds to the request and the web browser displays the response in the form of a website.

The Web Server layer contains components that parse the request (HTTP Request Parser) coming from the clients and forwards the response to them. It holds all the business logics and databases that are responsible for building websites and store data in them. Example: IIS Web Server, Apache Web Server, etc. In some cases, the users access the application through the presentation layer, which serves as an intermediary between the user and the Web Server. This layer includes the user interface components. The presentation layer is not an absolute requirement and the client layer can interact directly with the service layer.

The Business Layer is responsible for the core functioning of the system and includes business logic and applications, such as .NET that is used by the developers to build websites according to the clients' requirements. This layer also holds a legacy application, an older system integrated as an internal or external component.

The Database Layer comprises of cloud services, B2B layer that holds all the commercial transactions and a Database Server that supplies an organization's production data in a structured form. Example: MS SQL Server, MySQL server, etc.



Challenges in Web Application Forensics

01 Web applications are generally **distributed in nature**

02 Traces of activities are **recorded across a number** of hardware and software infrastructures

03 **Very limited or no downtime** is allowed for investigation

04 **Huge volume of logs** from different sources are analyzed and correlated

05 **Large databases** are analyzed

06 **Requires complete knowledge** of different web servers, application servers, databases and underlying applications

07 **Tracing back is difficult** in case of reverse proxies and anonymizers

Web applications serve a wide range of services and can support various types of servers like IIS, Apache, etc. Therefore, the forensic investigators must have good knowledge of various servers in order to examine the logs and understand them when an incident occurs.

Web applications are often business-critical, thus making it difficult for the investigators to create their forensic image that requires the site to be down for some time for completing the process. This makes it difficult for the investigators to capture volatile data including processes, port/network connections, logs of memory dumps, and user logs during the time of the incident analysis.

The investigators must have a good understanding of all kinds of web and applications servers in order to understand, analyze and correlate various formats of logs collected from their respective sources.

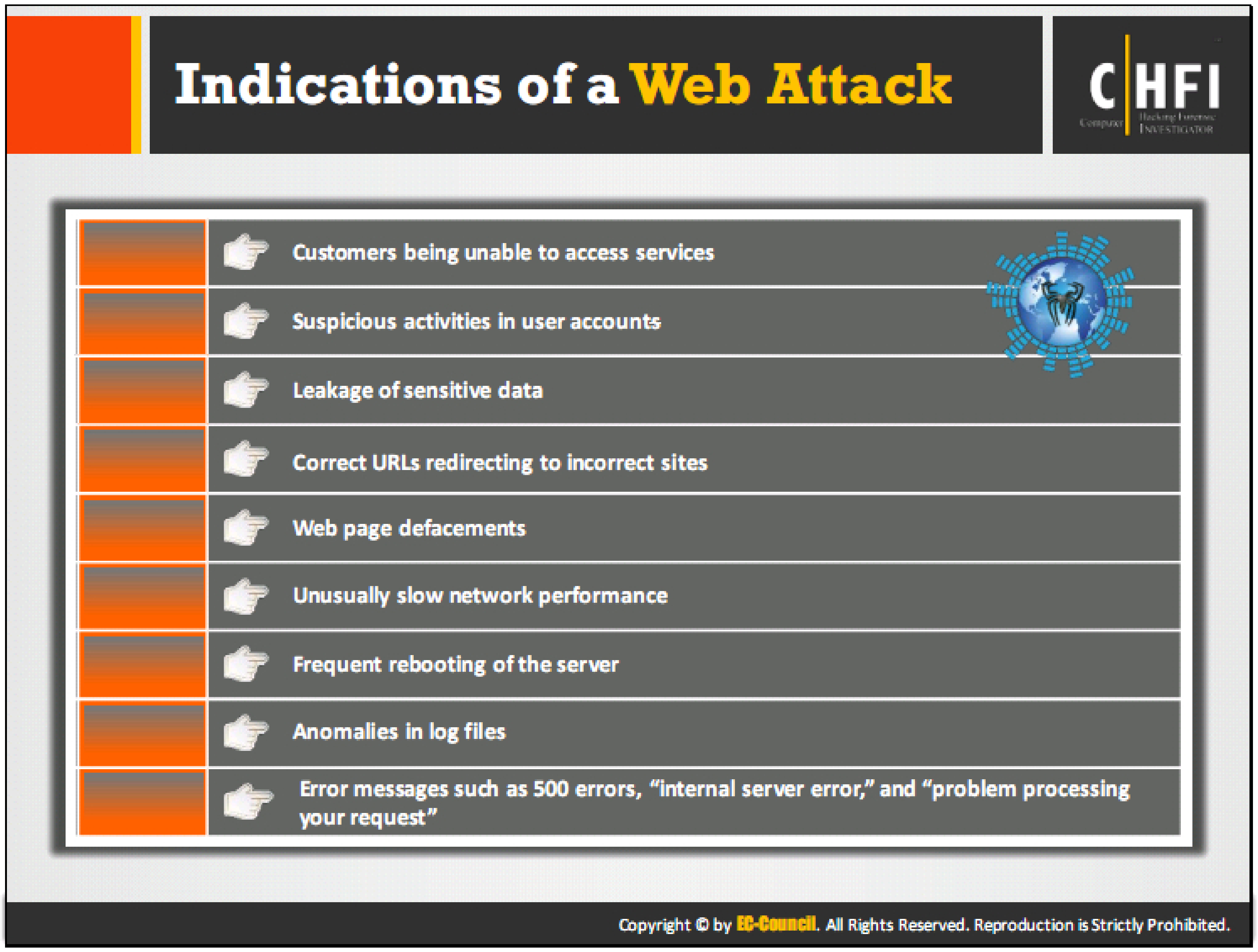
As the websites' traffic increases, the log files recorded in the database keeps on increasing. So, it becomes difficult for the investigators to collect and analyze these logs.

When a website attack occurs, the investigators need to gather the digital fingerprints left by the attacker. Then, they need to collect the following data fields associated with each HTTP request made to the website in order to get an insight of the attack performed.

- Date and time at which the request was sent
- IP Address from where the request has initiated
- HTTP method used (GET/POST)

- URI
- HTTP Query
- A full set of HTTP headers
- The Full HTTP Request body
- Event Logs (non-volatile data)
- File listings and timestamps (non-volatile data)

Most of the web applications restrict access to HTTP information, such as the full set of HTTP headers and the request body without which all the HTTP headers will look alike. This makes it impossible for the investigators to differentiate valid HTTP requests from the malicious ones.



Indications of a Web Attack

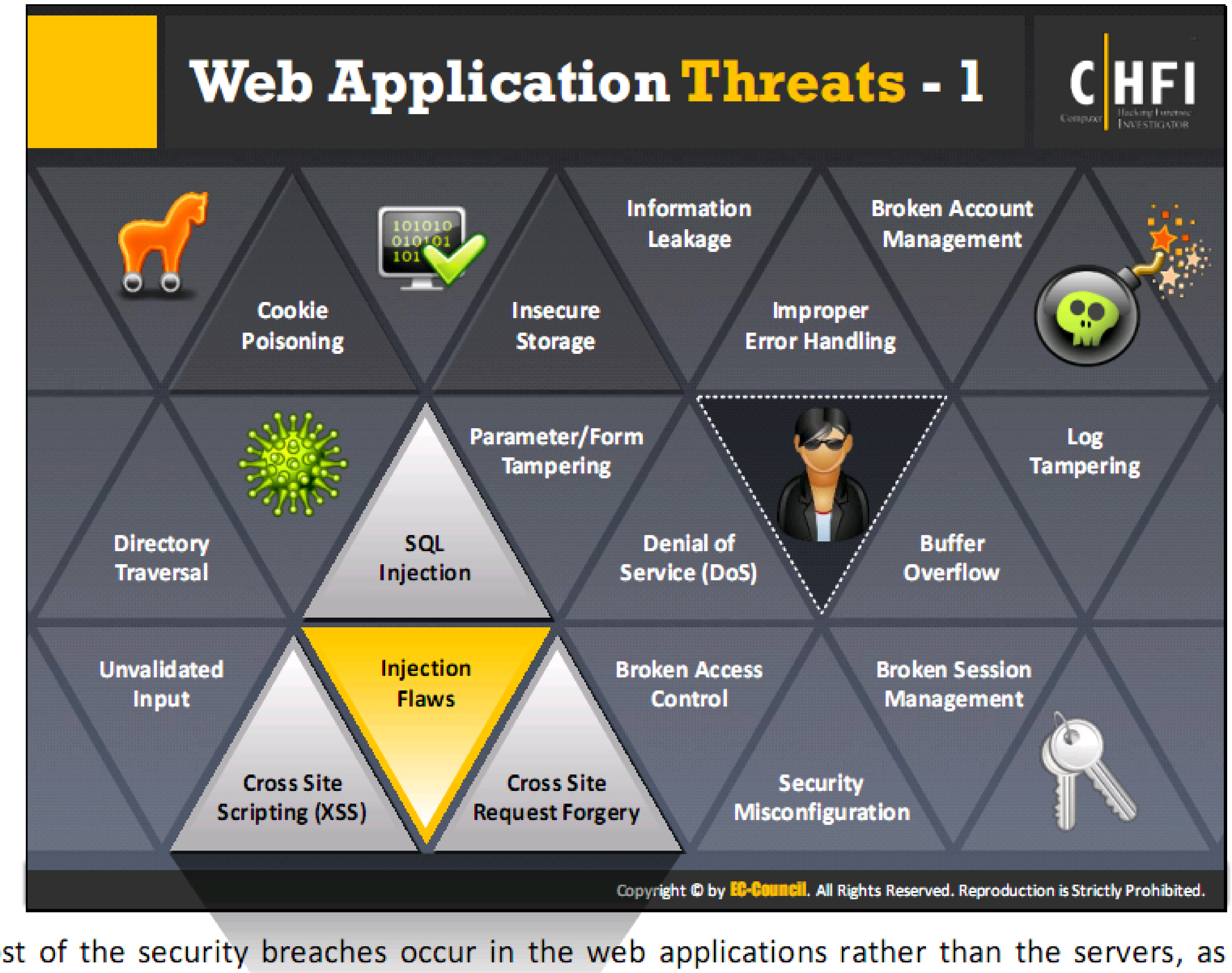
- Customers being unable to access services
- Suspicious activities in user accounts
- Leakage of sensitive data
- Correct URLs redirecting to incorrect sites
- Web page defacements
- Unusually slow network performance
- Frequent rebooting of the server
- Anomalies in log files
- Error messages such as 500 errors, "internal server error," and "problem processing your request"

There are different indications related to each type of attack. For example, in a denial of service (DoS) attack, the customers are denied any access to the information or services available on the website. In such cases, customers report the unavailability of online services because the attacker prevents the legitimate user from accessing websites, email accounts, and other services that rely on the victim's computer.

Another indication of a web attack can be redirecting of a web page (redirection attack – a common technique observed if an Exploit Kit is present on the web application) to an unknown website. When a user types the URL in the address bar, he or she is unable to access the site, and instead of accessing the typed site, the server redirects the user to some unknown site.

Unusual slow network performance and frequent rebooting of the server also gives an indication of a web attack.

Anomalies found in the log files are also an indication of web attacks. Change in the password and creation of a new user account also reveals the attack attempts. There may be other indications, such as the returning of error messages. For example, an HTTP 500 error message page indicates the occurrence of a SQL injection attack. There are other error messages, such as "an internal server error" or a "problem processing your request" that indicates a web attack.


Web Application Threats - 1
CHFI
Cookie Poisoning
Information Leakage
Broken Account Management
Insecure Storage
Improper Error Handling
Parameter/Form Tampering
Log Tampering
Directory Traversal
SQL Injection
Denial of Service (DoS)
Buffer Overflow
Unvalidated Input
Injection Flaws
Broken Access Control
Broken Session Management
Cross Site Scripting (XSS)
Cross Site Request Forgery
Security Misconfiguration
Copyright © by EC-Council. All Rights Reserved. Reproduction is Strictly Prohibited.

Most of the security breaches occur in the web applications rather than the servers, as web applications might contain insecure code (or bugs), which may be due to improper coding at the development phase. Due to this, the web applications are prone to various types of threats, few of which have been mentioned below:

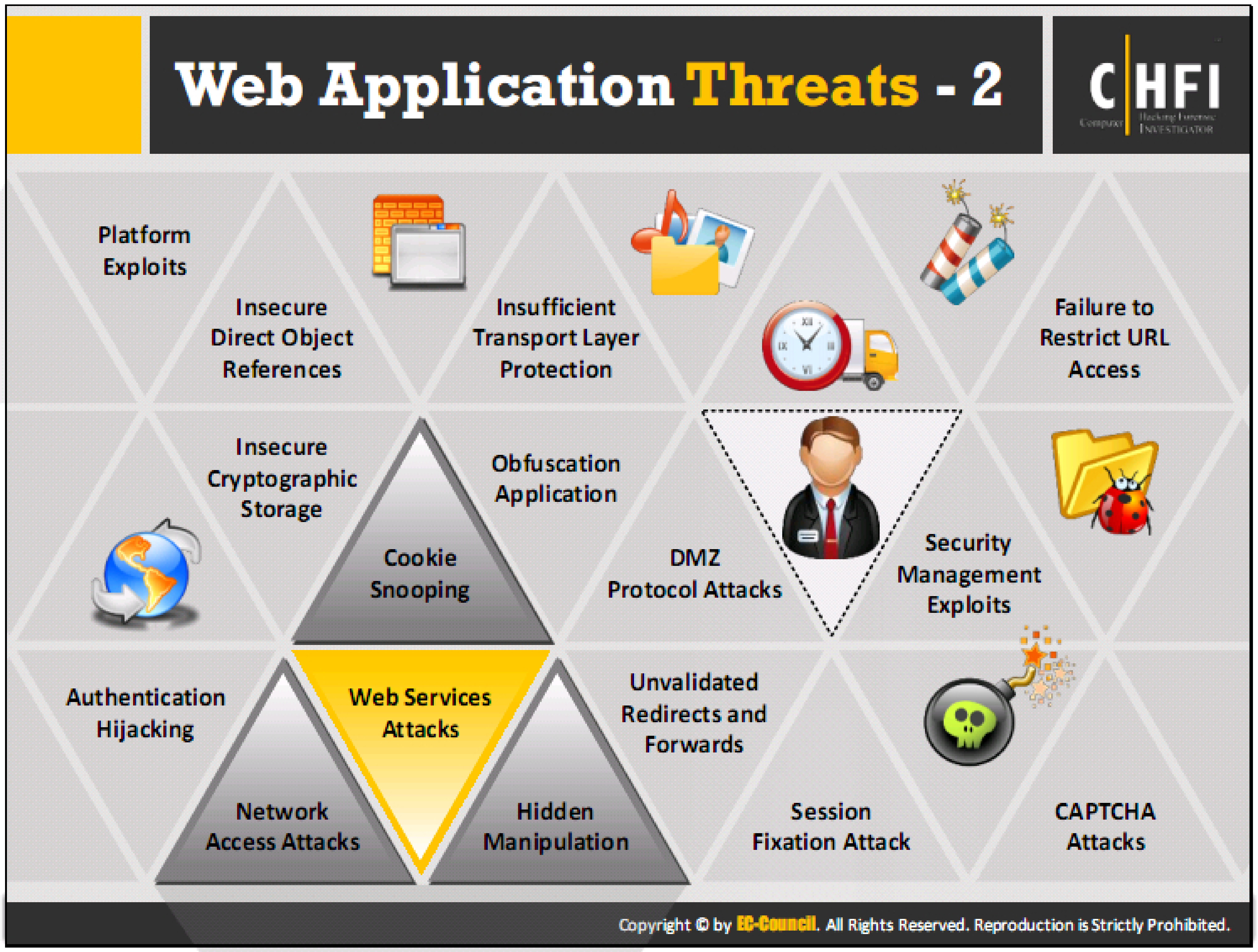
- **Buffer Overflow**: Buffer overflow vulnerability of a web application occurs when it fails to guard its buffer properly and allows writing beyond its maximum size. Thus, it overwrites the adjacent memory locations. There are multiple forms of buffer overflow, including a Heap Buffer Overflow and a Format String Attack. The purpose of these attacks is to corrupt the execution stack of the web application.

- **Cookie Poisoning:** Cookie Poisoning refers to the modification of a cookie for bypassing security measures or gaining unauthorized information. The attackers bypass the authentication process by altering the information present inside a cookie. Once the attackers gain control over a network, they can modify its content, use the system for a malicious attack, or steal information from the users' systems.

- **Insecure Storage:** The sensitive information, such as account records, credit card numbers, passwords or other authenticated information are generally stored by the web applications either in a database or on a file system. If the developers make any mistakes while enforcing the encryption techniques on a web application or ignore the security aspects of some parts of the application, this sensitive information might be at risk. Insecure storage of such data can allow the attacker to gain access to the web application as a legitimate user. Hence, the forensics investigators need to understand the process of storing the data.

- **Information Leakage:** Information leakage refers to a drawback in a web application where it unintentionally reveals the sensitive information to an unauthorized user. Such information leakage can cause great losses to any company. Hence, the company needs to employ proper content filtering mechanisms to protect all its information or data sources (such as systems or other network resources) from information leakage.
- **Improper Error Handling:** This threat arises when a web application is unable to handle internal errors properly. In such case, the website returns information, such as database dumps, stack traces, and error codes in the form of errors.
- **Broken Account Management:** It refers to vulnerable account management functions including account update, recovery of the forgotten or lost password or resetting the password, and such similar functions, which might weaken the valid authentication schemes.
- **Directory Traversal:** When attackers exploit HTTP by using directory traversal, they gain access to the unauthorized directories. Then, the attackers may execute commands outside the web server's root directory.
- **SQL Injection:** In this type of attack, the attacker injects SQL commands via input data. Later, the attacker is able to tamper with the data.
- **Parameter/Form Tampering:** This type of tampering attack intends at manipulating the communication parameters exchanged between the client and server to make changes in the application data, like user IDs and passwords with event logs, cost, and quantity of products, etc. In order to improve the functionality and control of the application, the system collects the information and stores in hidden form fields, cookies or URL query strings. Man in the middle is one of the examples of this type of attack. Hackers use tools like Webscarab and Paros proxy for the attacks.
- **Denial of Service (DoS):** The DoS attack is a method that intends at terminating the website operations or a server operation by making its resources unlivable to the clients. For example, a website related to banking or email service is not able to function for a few hours or even days, resulting in loss of both time and money.
- **Log Tampering:** Web applications maintain logs to track the usage patterns, such as admin login credentials and user login credentials. The attackers usually inject, delete or tamper the web application logs to engage in malicious activities or hide their identities.
- **Unvalidated Input:** In order to bypass the security system, the attackers tamper with the URL, HTTP requests, headers, hidden fields, form fields, query strings, etc. User login IDs and other related data get stored in the cookies and this becomes a source of attack. Examples of attacks that caused unvalidated input include SQL injection, cross-site scripting (XSS), buffer overflows, etc.

- **Cross Site Scripting:** The attackers bypass the client's ID security mechanisms and gain access privileges; and then inject the malicious scripts into specific fields in the web pages. These malicious scripts can even rewrite the HTML content of a website.
- **Injection Flaws:** The attackers inject malicious code, commands or scripts into the input gates of flawed web applications in such a way that the applications interpret and run with the newly supplied malicious input, which in turn allows them to extract sensitive information.
- **Cross Site Request Forgery:** In this attacking method, an authenticated user in made to perform certain tasks on the web application that is chosen by an attacker. Example: A user clicking on a particular link sent through an email or chat.
- **Broken Access Control:** This is a method in which an attacker identifies a flaw related to access control and bypasses the authentication, and then compromises the network.


Web Application Threats - 2
CHFI
Platform Exploits
Insecure Direct Object References
Insufficient Transport Layer Protection
Failure to Restrict URL Access
Insecure Cryptographic Storage
Obfuscation Application
Cookie Snooping
DMZ Protocol Attacks
Security Management Exploits
Authentication Hijacking
Web Services Attacks
Unvalidated Redirects and Forwards
Network Access Attacks
Hidden Manipulation
Session Fixation Attack
CAPTCHA Attacks
Copyright © by EC-Council. All Rights Reserved. Reproduction is Strictly Prohibited.

Discussed below are a few more types of web application threats:

- **Platform Exploits:** The web developers use specific application platforms, for instance, Microsoft .Net, Sun Java technologies, IBM Websphere, etc., to develop web applications. These platforms may contain vulnerabilities, such as application misconfiguration, bugs, etc., which might act as attack vectors for exploiting the web applications.

- **Insecure Direct Object References:** When developers expose various internal implementation objects such as files, directories, database records, or key-through references, it results in an insecure direct object reference. For example, if a bank account number is a primary key, there is a chance of attackers compromising the application and taking advantage of such references.

- **Insufficient Transport Layer Protection:** The developers need to enforce SSL/TLS security technology for the website authentication. Failing to implement, attackers can access session cookies by monitoring the network flow. Various threats such as phishing attacks, account theft, and admin account creation may occur after gaining the cookies.

- **SSL/TLS Downgrade Attack:** All major browsers are susceptible to protocol downgrade attacks; an active MITM can simulate failure conditions and force all browsers to downgrade from attempting to negotiate TLS 1.2, making them fall back to SSL 3. At that point, a cryptographic attack can occur (see POODLE attack); however, it requires MTiM access.

- **Failure to Restrict URL Access:** An application often safeguards or protects sensitive functionality and prevents the display of links or URLs for protection. Failure to Restrict URL Access refers to the vulnerability where a web application is unable to restrict a hacker from accessing a particular URL. Here, an attacker tries to bypass the website security using techniques, such as forced browsing and gains unauthorized access to specific web pages or other data files containing sensitive information.

- **Insecure or Improper Cryptographic Storage:** The sensitive data stored in a database should be properly encrypted using cryptography. However, some cryptographic encryption methods contain inherent vulnerabilities. Therefore, the developers should use strong encryption methods to develop secure applications. In addition, they must securely store the cryptographic keys, so that the attackers cannot easily obtain them and decrypt the sensitive data.

- **Cookie Snooping:** An attacker using a local proxy decodes or cracks user credentials. Once the attacker gains these plain text credentials, he/she logs into the system as a legitimate user and gains access to unauthorized information.

- **Obfuscation Application:** Obfuscation is a technique used by the attackers to create a number of variants of malicious code, thereby making it difficult for security mechanisms, such as web application firewalls, intrusion detection systems, etc., to detect it.

- **Demilitarized Zone (DMZ) Protocol Attacks:** The DMZ is a semi-trusted network zone that separates the untrusted Internet from the company's trusted internal network. An attacker who is able to compromise a system that allows other DMZ protocols, also gets access to other DMZ and internal systems. This can further lead to:
  - Web application and data compromise
  - Website defacement
  - Access to internal systems that includes backups, databases and source code

- **Security Management Exploits:** Some attackers target security management systems, either on networks or on the application layer, in order to modify or disable security enforcement. An attacker who exploits security management can directly modify protection policies, delete existing policies, add new policies, and modify applications, system data, and resources.

- **Authentication Hijacking:** All web applications rely on information, such as password and User ID, for user identification. The attackers try to hijack those credentials using various attack techniques like sniffing, social engineering, etc. Once they obtain these credentials, they perform various malicious acts, including session hijacking, service theft, and user impersonation.

- **Network Access Attacks:** These attacks can majorly affect the web applications, including the basic level of service. They can also allow levels of access that the standard HTTP application methods could not grant.

- **Web Services Attacks:** The attacker can get into the target web applications by exploiting an application integrated with vulnerable web services. An attacker injects a malicious script into a web service and is able to disclose and modify application data.

- **Hidden Manipulation:** The attackers attempting to compromise the e-commerce websites mostly use these types of attacks. They manipulate the hidden fields and change the data stored in them. They can substitute the original prices with the price of their choice and conclude the transactions. This sort of attack is faced by many online stores.

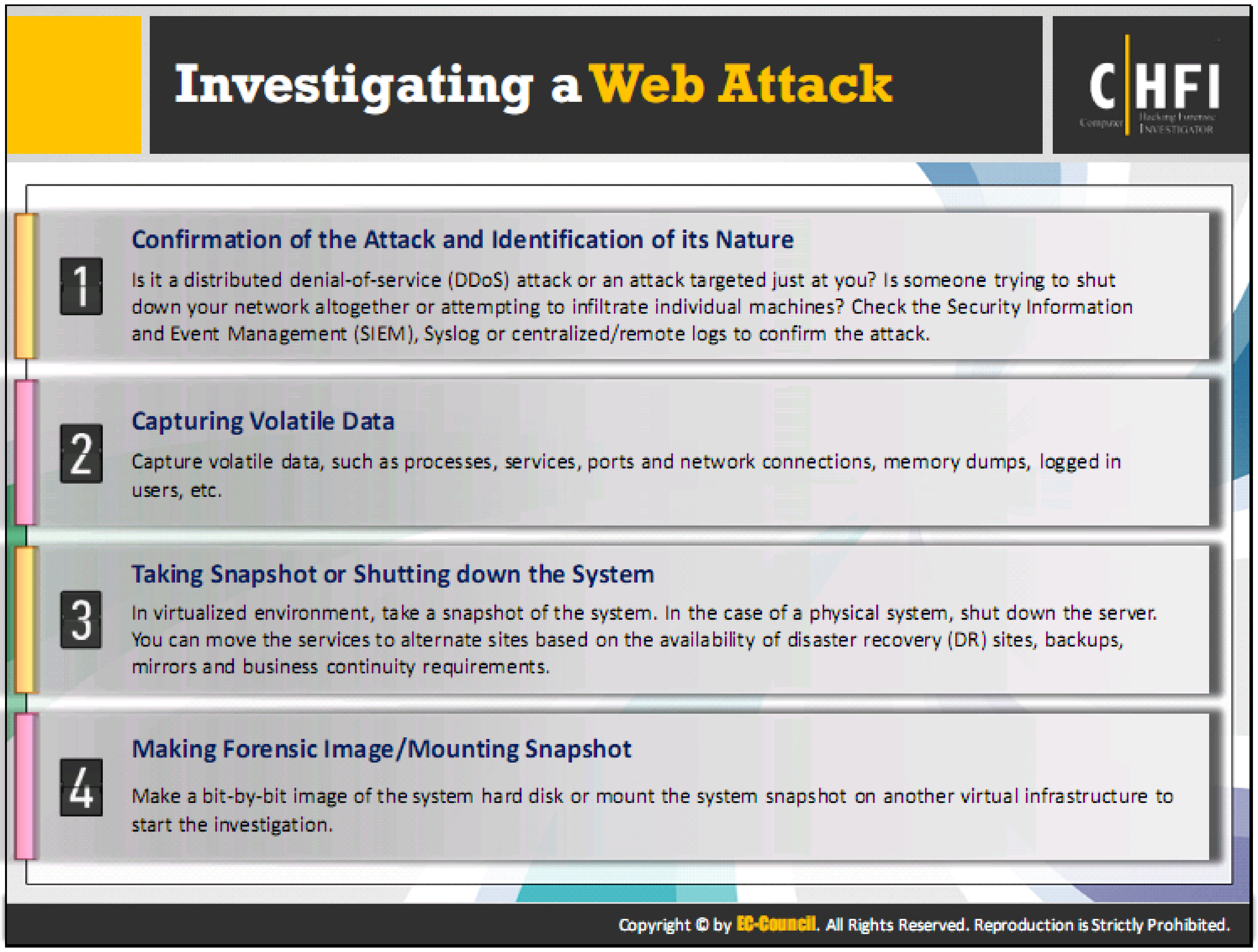
- **Unvalidated Redirects and Forwards:** The attackers lure the victim and make them click on the unvalidated links that appear to be legitimate. Such redirects may lead to the installation of malware or trick the victims to share their passwords or other sensitive information. Such unsafe forwards may lead to access control bypass, further resulting in:

  - Session fixation attacks
  - Security management exploits
  - Failure to restrict URL access
  - Malicious file execution

- **Session Fixation Attack:** This type of attack assists the attacker in hijacking a valid user session. The attacker hijacks the user-validated session with prior knowledge of the user ID session, by authenticating with a known session ID. In this attack-type, the attacker tricks the user to access a genuine web server using an explicit session ID value. The attacker assumes the identity of the victim and exploits those credentials at the server. The steps involved are as follows:

  1. The attacker visits the bank website and logs in using his credentials.
  2. The web server sets a session ID on the attacker's machine.
  3. The attacker sends an email containing a link with a fixed session ID.
  4. The user clicks the link and is redirected to the bank website.
  5. The user logs in to the server using his credentials and fixed session ID.
  6. The attacker logs into the server using the victim's credentials with the same session ID.

- **CAPTCHA Attacks:** Implementing Completely Automated Public Turing test to tell Computers and Humans Apart (CAPTCHAs) prevents the automated software from performing actions that degrade the quality of service of a given system, which may be due to abuse or resource expenditure. CAPTCHAs aim at ensuring that the users of applications are human and ultimately aid in preventing unauthorized access and abuse. Each CAPTCHA implementation derives its strength by increasing the system's complexity to perform segmentation, image preprocessing, and classification.



Investigating a Web Attack

1 **Confirmation of the Attack and Identification of its Nature**

Is it a distributed denial-of-service (DDoS) attack or an attack targeted just at you? Is someone trying to shut down your network altogether or attempting to infiltrate individual machines? Check the Security Information and Event Management (SIEM), Syslog or centralized/remote logs to confirm the attack.

2 **Capturing Volatile Data**

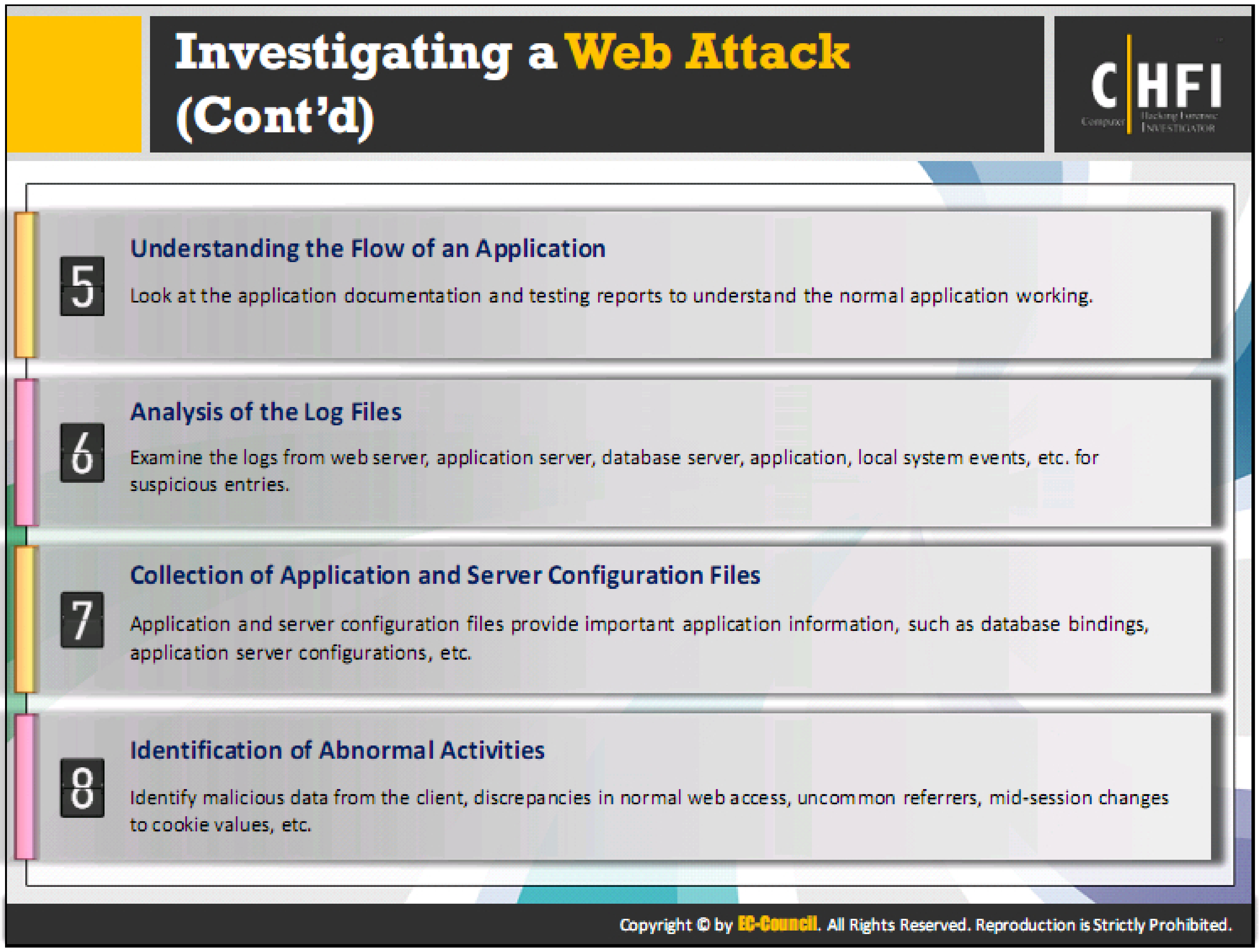
Capture volatile data, such as processes, services, ports and network connections, memory dumps, logged in users, etc.

3 **Taking Snapshot or Shutting down the System**

In virtualized environment, take a snapshot of the system. In the case of a physical system, shut down the server. You can move the services to alternate sites based on the availability of disaster recovery (DR) sites, backups, mirrors and business continuity requirements.

4 **Making Forensic Image/Mounting Snapshot**

Make a bit-by-bit image of the system hard disk or mount the system snapshot on another virtual infrastructure to start the investigation.



Investigating a Web Attack (Cont'd)

5 **Understanding the Flow of an Application**

Look at the application documentation and testing reports to understand the normal application working.

6 **Analysis of the Log Files**

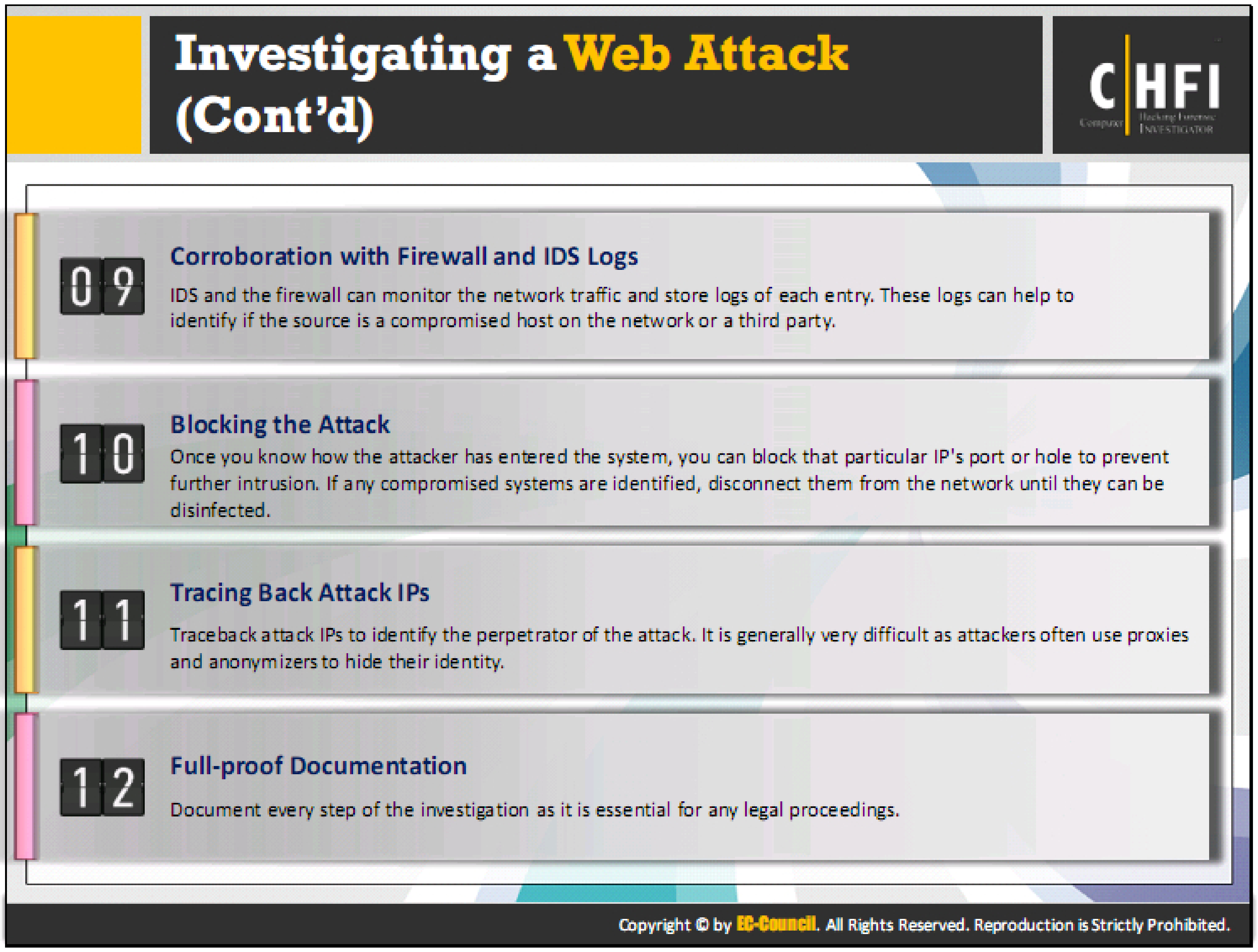
Examine the logs from web server, application server, database server, application, local system events, etc. for suspicious entries.

7 **Collection of Application and Server Configuration Files**

Application and server configuration files provide important application information, such as database bindings, application server configurations, etc.

8 **Identification of Abnormal Activities**

Identify malicious data from the client, discrepancies in normal web access, uncommon referrers, mid-session changes to cookie values, etc.



Investigating a Web Attack (Cont'd)

09 **Corroboration with Firewall and IDS Logs**

IDS and the firewall can monitor the network traffic and store logs of each entry. These logs can help to identify if the source is a compromised host on the network or a third party.

10 **Blocking the Attack**

Once you know how the attacker has entered the system, you can block that particular IP's port or hole to prevent further intrusion. If any compromised systems are identified, disconnect them from the network until they can be disinfected.

11 **Tracing Back Attack IPs**

Traceback attack IPs to identify the perpetrator of the attack. It is generally very difficult as attackers often use proxies and anonymizers to hide their identity.

12 **Full-proof Documentation**

Document every step of the investigation as it is essential for any legal proceedings.

Web applications have become a primary source of information exchange and management, in various enterprises, government agencies, etc. Because of their wide usage, web applications are becoming the primary targets for attackers. Information security professionals implement specific security measures to detect or prevent the attacks, but they cannot trace these attacks; allowing attackers to attempt new attacks on the target. This is where forensic investigation helps mitigate the attacks occurring on the application.

Forensic investigators examine the affected application and trace the attack signatures. This result in decrease in the number of attacks targeting the application, thereby, improving its security.

The steps involved in an investigation of web attacks are discussed in the above slide.

Investigating Web Attacks in Windows-Based Servers

Run **Event Viewer** to look at the logs:

```
C:\> eventvwr.msc
```

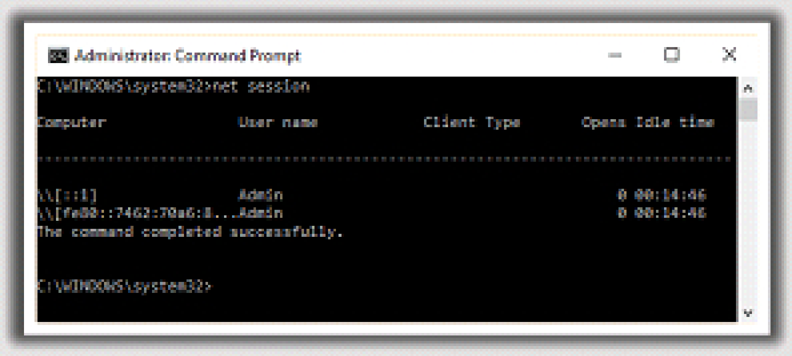
Check if the following **suspicious events** have occurred:

- Event log service ends
- Windows File Protection is inactive on the system
- The MS Telnet Service is running

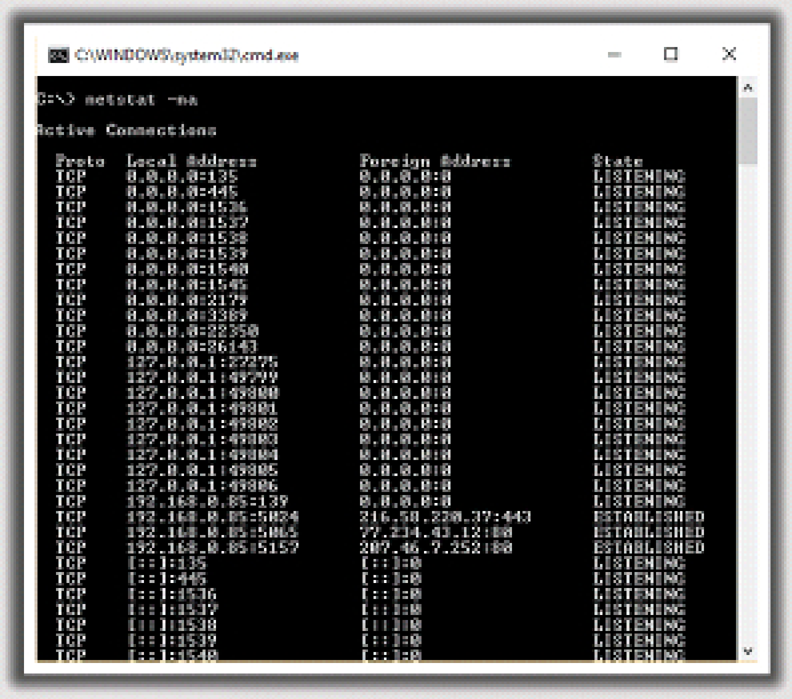
Find if the system has **failed login** attempts or **locked-out accounts**

Investigating Web Attacks in Windows-Based Servers (Cont'd)

- Review file shares to ensure their purpose
  `C:\> net view <IP Address>`
- Verify the users using open sessions
  `C:\> net session`
- Check if the sessions have been opened with other systems
  `C:\> net use`
- Analyze at NetBIOS over TCP/IP activity
  `C:\> nbtstat -S`
- Find if TCP and UDP ports have unusual listening
  `C:\> netstat -na`

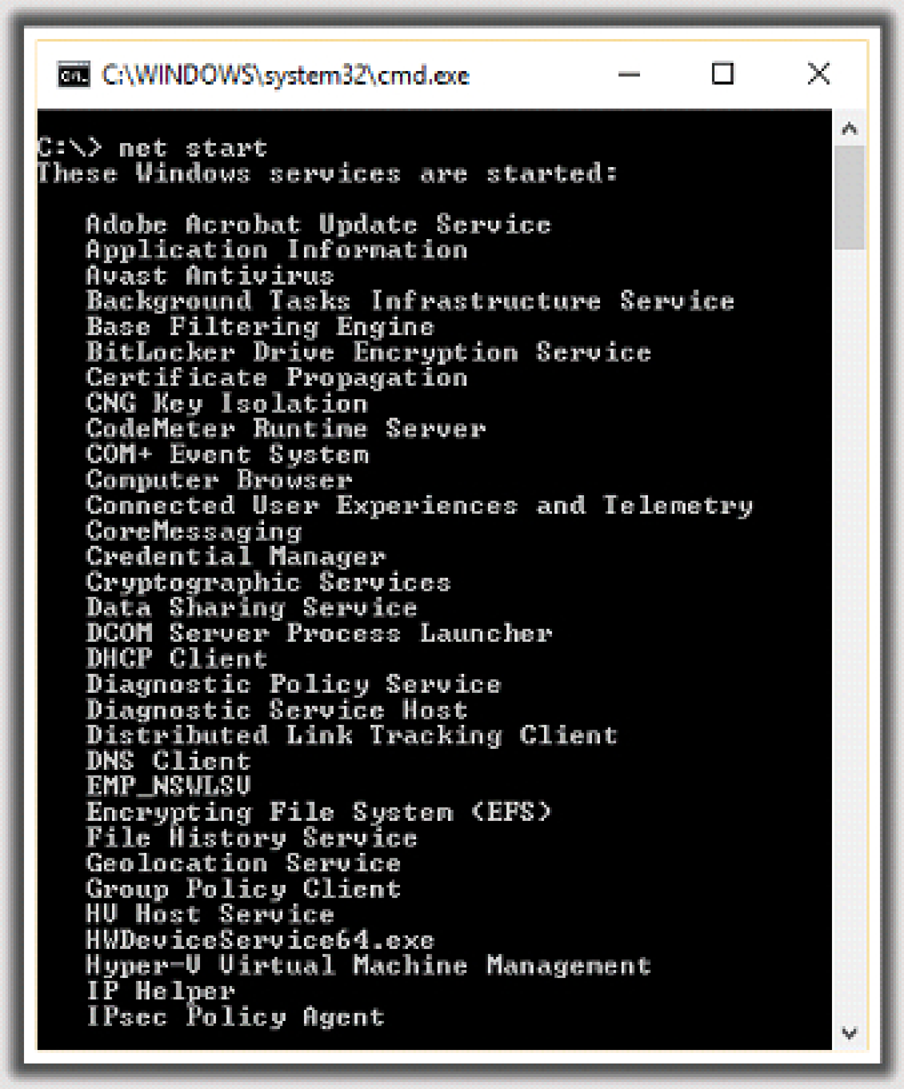


```
C:\WINDOWS\system32>net session

Computer               User name            Client Type       Opens Idle time

-------------------------------------------------------------------------------
\\[::1]                Admin                                      0 00:14:46
\\[fe80::7462:70a6:8...Admin                                      0 00:14:46
The command completed successfully.


C:\WINDOWS\system32>
```



```
C:\> netstat -na

Active Connections

  Proto  Local Address          Foreign Address        State
  TCP    0.0.0.0:135            0.0.0.0:0              LISTENING
  TCP    0.0.0.0:445            0.0.0.0:0              LISTENING
  TCP    0.0.0.0:1536           0.0.0.0:0              LISTENING
  TCP    0.0.0.0:1537           0.0.0.0:0              LISTENING
  TCP    0.0.0.0:1538           0.0.0.0:0              LISTENING
  TCP    0.0.0.0:1539           0.0.0.0:0              LISTENING
  TCP    0.0.0.0:1540           0.0.0.0:0              LISTENING
  TCP    0.0.0.0:1545           0.0.0.0:0              LISTENING
  TCP    0.0.0.0:2179           0.0.0.0:0              LISTENING
  TCP    0.0.0.0:3389           0.0.0.0:0              LISTENING
  TCP    0.0.0.0:22350          0.0.0.0:0              LISTENING
  TCP    0.0.0.0:26143          0.0.0.0:0              LISTENING
  TCP    127.0.0.1:27275        0.0.0.0:0              LISTENING
  TCP    127.0.0.1:49799        0.0.0.0:0              LISTENING
  TCP    127.0.0.1:49800        0.0.0.0:0              LISTENING
  TCP    127.0.0.1:49801        0.0.0.0:0              LISTENING
  TCP    127.0.0.1:49802        0.0.0.0:0              LISTENING
  TCP    127.0.0.1:49803        0.0.0.0:0              LISTENING
  TCP    127.0.0.1:49804        0.0.0.0:0              LISTENING
  TCP    127.0.0.1:49805        0.0.0.0:0              LISTENING
  TCP    127.0.0.1:49806        0.0.0.0:0              LISTENING
  TCP    192.168.0.85:139       0.0.0.0:0              LISTENING
  TCP    192.168.0.85:5024      216.58.220.37:443      ESTABLISHED
  TCP    192.168.0.85:5065      77.234.43.12:80        ESTABLISHED
  TCP    192.168.0.85:5157      207.46.7.252:80        ESTABLISHED
  TCP    [::]:135               [::]:0                 LISTENING
  TCP    [::]:445               [::]:0                 LISTENING
  TCP    [::]:1536              [::]:0                 LISTENING
  TCP    [::]:1537              [::]:0                 LISTENING
  TCP    [::]:1538              [::]:0                 LISTENING
  TCP    [::]:1539              [::]:0                 LISTENING
  TCP    [::]:1540              [::]:0                 LISTENING
```

Investigating Web Attacks in Windows-Based Servers (Cont'd)



```
C:\> net start
These Windows services are started:

    Adobe Acrobat Update Service
    Application Information
    Avast Antivirus
    Background Tasks Infrastructure Service
    Base Filtering Engine
    BitLocker Drive Encryption Service
    Certificate Propagation
    CNG Key Isolation
    CodeMeter Runtime Server
    COM+ Event System
    Computer Browser
    Connected User Experiences and Telemetry
    CoreMessaging
    Credential Manager
    Cryptographic Services
    Data Sharing Service
    DCOM Server Process Launcher
    DHCP Client
    Diagnostic Policy Service
    Diagnostic Service Host
    Distributed Link Tracking Client
    DNS Client
    EMP_NSWLSV
    Encrypting File System (EFS)
    File History Service
    Geolocation Service
    Group Policy Client
    HV Host Service
    HWDeviceService64.exe
    Hyper-V Virtual Machine Management
    IP Helper
    IPsec Policy Agent
```

Find scheduled and unscheduled tasks on the local host

`C:\> schtasks.exe`

Check for creation of new accounts in administrator group

`C:\> lusrmgr.msc`

See if any unexpected processes are running in Task Manager

`Start -> Run -> taskmgr -> OK`

Look for unusual network services

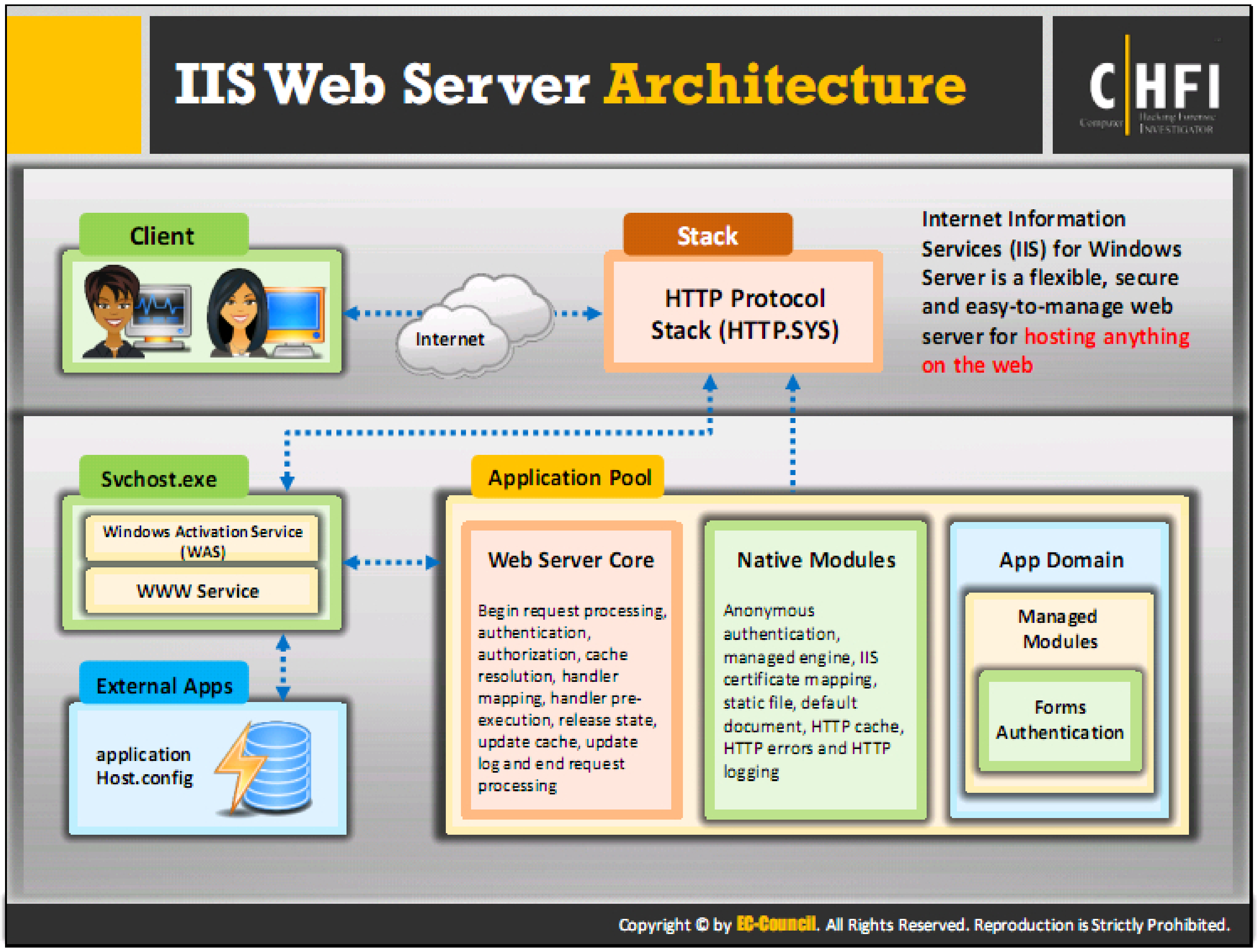
`C:\> net start`

Check file space usage to look for a sudden decrease in free space

`C:\> dir`

Microsoft Windows-based operating systems constitute 89.34% of the market share according to www.netmarketshare.com, which means that the developers might prefer to use Windows-based servers to deploy web applications compared to other operating systems. Due to their wide usage, these operating systems and web applications hosted in some of these operating systems become a primary target for the attackers. The attackers may attempt to either exploit the vulnerabilities contained in the Windows-based server or the web applications and gain unauthorized access to their resources.

When an attack occurs on a web application, the investigators examine the attack on the server hosting the web application by using some of the inbuilt tools and applications of Windows-based machines as shown above.



IIS Web Server Architecture

Internet Information Services (IIS) for Windows Server is a flexible, secure and easy-to-manage web server for hosting anything on the web

CHFI
Client
Internet
Stack
HTTP Protocol Stack (HTTP.SYS)
Svchost.exe
Windows Activation Service (WAS)
WWW Service
External Apps
application Host.config
Application Pool
Web Server Core
Begin request processing, authentication, authorization, cache resolution, handler mapping, handler pre-execution, release state, update cache, update log and end request processing
Native Modules
Anonymous authentication, managed engine, IIS certificate mapping, static file, default document, HTTP cache, HTTP errors and HTTP logging
App Domain
Managed Modules
Forms Authentication

Internet Information Server (IIS), a Microsoft-developed application, is a Visual Basic code application that lives on a Web server and responds to requests from the browser. It supports HTTP, HTTPS, FTP, FTPS, SMTP, and NNTP. An IIS application uses HTML to present its user interface and uses compiled Visual Basic code to process the requests and respond to events in the browser. IIS for Windows Server is a flexible and easy-to-manage Web server for web hosting.

The IIS server constitutes 29.83% of the market share according to https://news.netcraft.com, February 2016.

IIS provides various components with important functionality for the application and web server roles in Windows Server machines.

IIS components include:

- Protocol listeners (HTTP.sys)
- Web services like World Wide Web Publishing Service (WWW service)
- Windows Process Activation Service (WAS)

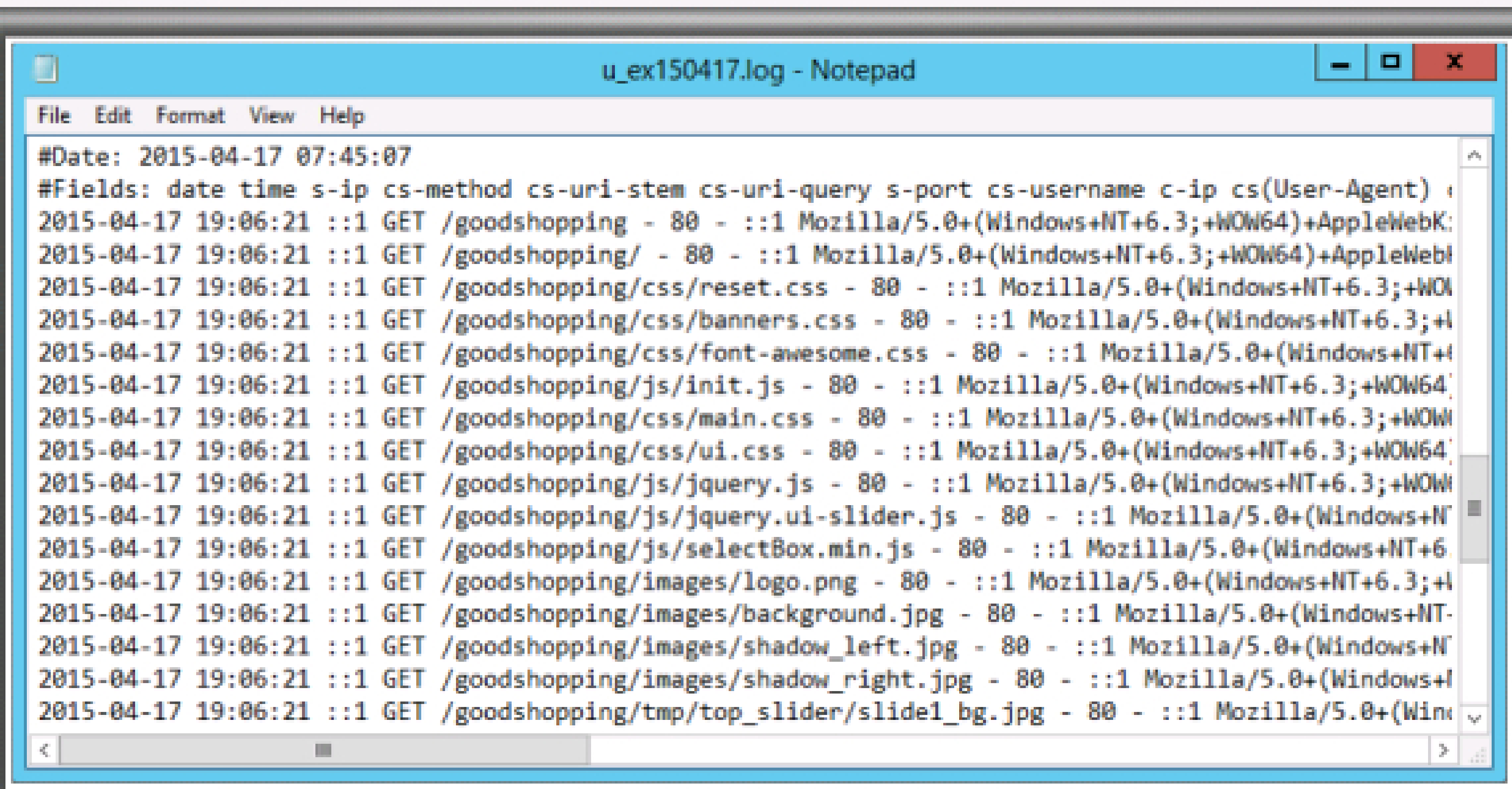
IIS components' responsibilities include:

- Listening to the requests coming from the server
- Managing processes
- Reading configuration files

IIS depends mostly on a group of dynamic-link libraries (DLLs) that work collectively with the main server process (inetinfo.exe) capturing different functions, for e.g., content indexing, server-side scripting, web- based printing, etc. The open architecture of IIS enables an attacker to exploit the web with malicious content. Without service packs or hot fixes in IIS web server, there are numerous possibilities that an IIS process inetinfo.exe calls a command shell. This is disturbing, as there is no inherent need for inetinfo.exe to invoke a command prompt.

IIS Logs

- IIS logs all server visits in log files
- IIS logs provide useful information regarding the activity of various Web applications, such as connection time, IP address, user account, page URLs, and actions
- The IIS server generates ASCII text-based log files
- On Windows Server 2012, the log files are stored by default in the *%SystemDrive%\inetpub\logs\LogFiles*



```
#Date: 2015-04-17 07:45:07
#Fields: date time s-ip cs-method cs-uri-stem cs-uri-query s-port cs-username c-ip cs(User-Agent)
2015-04-17 19:06:21 ::1 GET /goodshopping - 80 - ::1 Mozilla/5.0+(Windows+NT+6.3;+WOW64)+AppleWebK
2015-04-17 19:06:21 ::1 GET /goodshopping/ - 80 - ::1 Mozilla/5.0+(Windows+NT+6.3;+WOW64)+AppleWeb
2015-04-17 19:06:21 ::1 GET /goodshopping/css/reset.css - 80 - ::1 Mozilla/5.0+(Windows+NT+6.3;+WO
2015-04-17 19:06:21 ::1 GET /goodshopping/css/banners.css - 80 - ::1 Mozilla/5.0+(Windows+NT+6.3;+
2015-04-17 19:06:21 ::1 GET /goodshopping/css/font-awesome.css - 80 - ::1 Mozilla/5.0+(Windows+NT+
2015-04-17 19:06:21 ::1 GET /goodshopping/js/init.js - 80 - ::1 Mozilla/5.0+(Windows+NT+6.3;+WOW64
2015-04-17 19:06:21 ::1 GET /goodshopping/css/main.css - 80 - ::1 Mozilla/5.0+(Windows+NT+6.3;+WOW
2015-04-17 19:06:21 ::1 GET /goodshopping/css/ui.css - 80 - ::1 Mozilla/5.0+(Windows+NT+6.3;+WOW64
2015-04-17 19:06:21 ::1 GET /goodshopping/js/jquery.js - 80 - ::1 Mozilla/5.0+(Windows+NT+6.3;+WOW
2015-04-17 19:06:21 ::1 GET /goodshopping/js/jquery.ui-slider.js - 80 - ::1 Mozilla/5.0+(Windows+NT
2015-04-17 19:06:21 ::1 GET /goodshopping/js/selectBox.min.js - 80 - ::1 Mozilla/5.0+(Windows+NT+6
2015-04-17 19:06:21 ::1 GET /goodshopping/images/logo.png - 80 - ::1 Mozilla/5.0+(Windows+NT+6.3;+
2015-04-17 19:06:21 ::1 GET /goodshopping/images/background.jpg - 80 - ::1 Mozilla/5.0+(Windows+NT-
2015-04-17 19:06:21 ::1 GET /goodshopping/images/shadow_left.jpg - 80 - ::1 Mozilla/5.0+(Windows+NT
2015-04-17 19:06:21 ::1 GET /goodshopping/images/shadow_right.jpg - 80 - ::1 Mozilla/5.0+(Windows+
2015-04-17 19:06:21 ::1 GET /goodshopping/tmp/top_slider/slide1_bg.jpg - 80 - ::1 Mozilla/5.0+(Win
```

Investigating IIS Logs

- **Example of IIS log file entry as viewed in a text editor:**

  2016-02-10 06:11:41 192.168.0.10 GET /images/content/bg_body_1.jpg - 80 - 192.168.0.27 Mozilla/5.0+(Windows+NT+6.3;+WOW64)+AppleWebKit/537.36+(KHTML,+like+Gecko)+Chrome/48.0.2564.103 +Safari/537.36 http://www.moviescope.com/css/style.css 200 0 0 365

Field	Appear As	Description
Date	03/06/2015	Log file entry was made on June 03, 2015
Time	8:45:30	Log life entry was recorded at 8:45 A.M
Server IP	172.15.10.30	IP address of the server
Client IP address	192.168.100.150	IP address of the client
cs-method	GET	The user issued a GET or download command
cs-uri-stem	/images/content/bg_body_1.jpg	The user wanted to download the bg_body_1.jpg file from the Images folder
cs-uri-query	-	The URI query did not occur (URI queries are necessary only for dynamic pages, such as ASP pages, so this field usually contains a hyphen for static pages.)
s-port	80	The server port
cs-username	-	The user was anonymous
c-ip	192.168.0.27	The IP address of the client
cs(User-Agent)	Mozilla/5.0+(Windows+NT+6.3;+WOW64)+AppleWebKit/537.36+(KHTML,+like+Gecko)+Chrome/48.0.2564.103+Safari/537.36	The type of browser that the client used, as represented by the browser
cs(Referer)	http://www.moviescope.com/css/style.css	The Web page that provided the link to the Web site
sc-status	200	The request was fulfilled without error
time-taken	365	The action was completed in 365 milliseconds

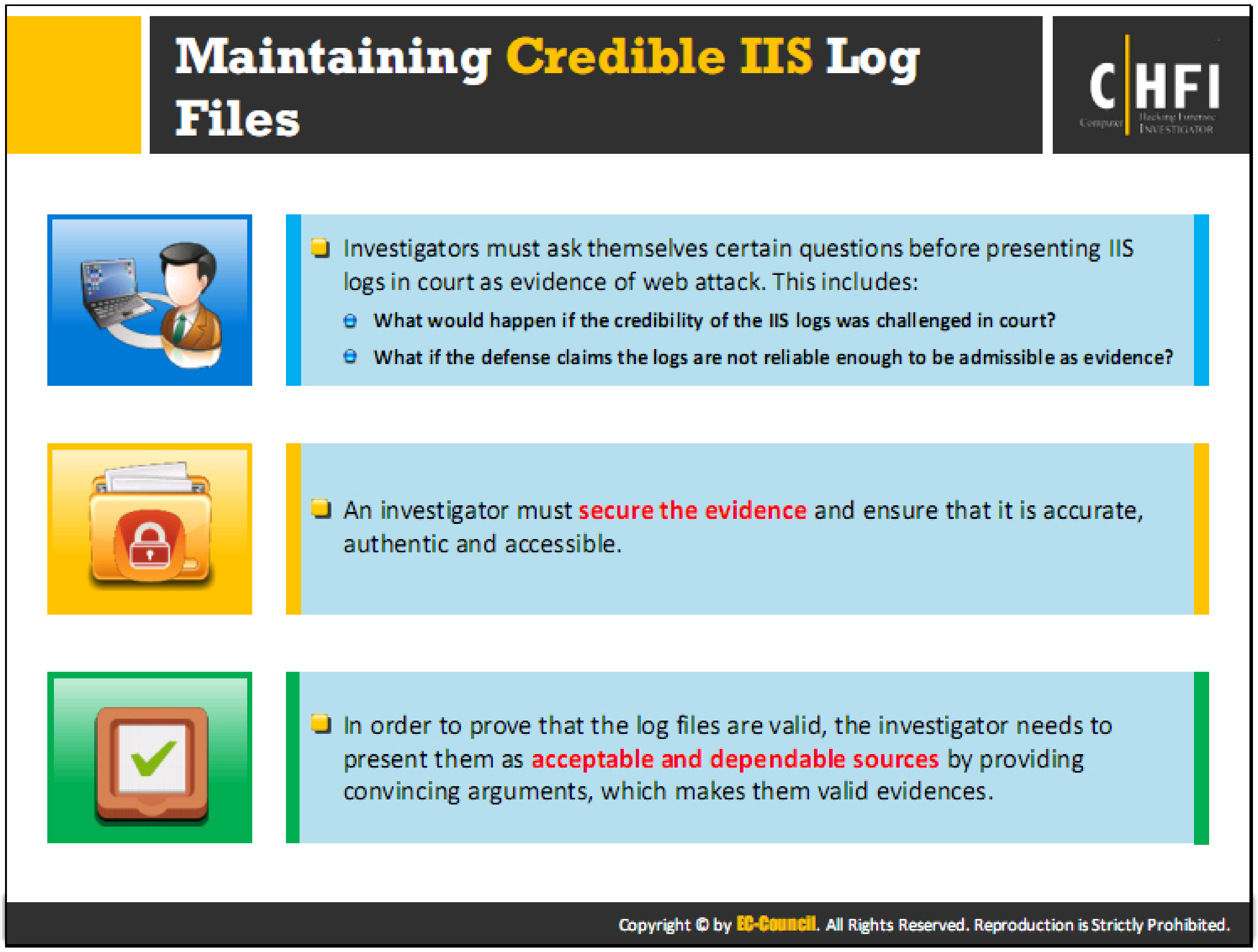
The IIS server might become vulnerable if there are any coding or configuration issues, which can allow attackers to exploit it if not addressed on time. On the occurrence of such attacks, forensic investigators examine the IIS logs to trace the attempts made by the attacker to exploit the server. The IIS logs provide useful information regarding the user activities. Most often, the log file(s) is/are located at **%SystemDrive%\inetpub\logs\LogFiles**.

Note: The log storage location may vary if the administrator has made a configuration to record and store the logs in some other location. However, in general, From the Windows Start menu, go to Administrative Tools and click on Internet Information Services (IIS) Manager. Expand the server name's folder and click on the Sites folder to load a list of sites in the content pane. Open its settings in the content pane. (Alternatively, you can expand the Sites folder and click on the site name in the left hand tree view.) Select Logging from the content pane to load the Logging settings. In the Directory field, you'll find the path in which your logs reside. Navigate to the LogFiles folder by following the path contained in the Directory field.

Within the LogFiles folder you'll find a subfolder for each site configured in labeled as W3SVC1, W3SVC2, etc. The last number in the folder name corresponds to the SiteID. Find the folder that matches the site's ID.

Each virtual server has its own subdirectory for log files, named **W3SVCn**, where '**n**' represents the number of the virtual server. The W3SVCn subdirectories store log files named **u_exyymmdd.log**, where '**yy**' refers to the year, '**mm**' refers to a month, and '**dd**' refers to the date.

IIS log file is a non-customized or fixed ASCII text based format. The IIS format includes basic items, such as client IP address, username, date and time, service and instance, server name and IP address, request type, target of operation, etc.



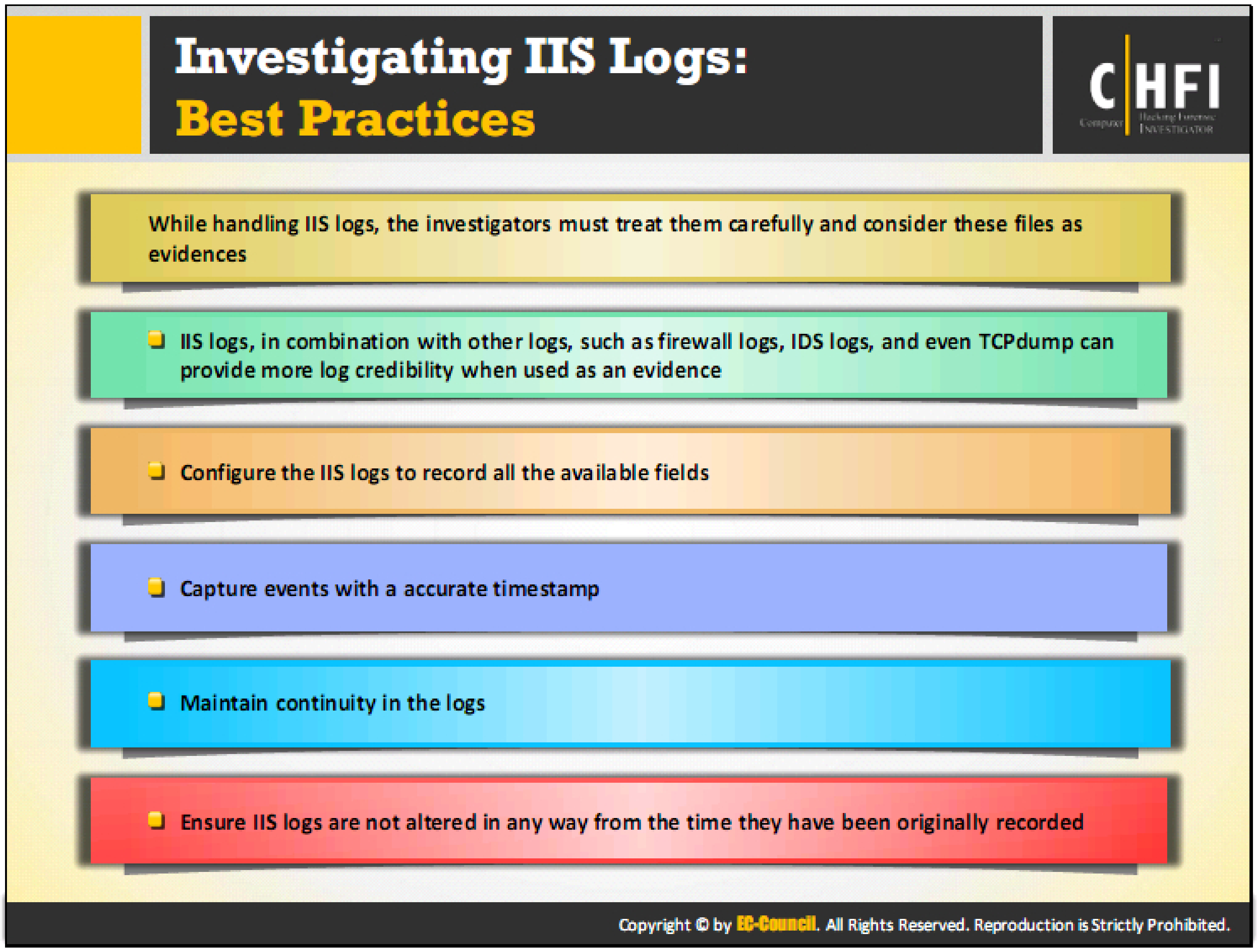
Maintaining Credible IIS Log Files

- Investigators must ask themselves certain questions before presenting IIS logs in court as evidence of web attack. This includes:
  - What would happen if the credibility of the IIS logs was challenged in court?
  - What if the defense claims the logs are not reliable enough to be admissible as evidence?

- An investigator must secure the evidence and ensure that it is accurate, authentic and accessible.

- In order to prove that the log files are valid, the investigator needs to present them as acceptable and dependable sources by providing convincing arguments, which makes them valid evidences.

It is very crucial to maintain the credibility of the IIS log files as they are the principle evidence used by the forensic investigators to investigate web attacks. Before presenting the evidence in the court, it is essential to present convincing arguments to prove that the submitted evidence (log files) is trustworthy and substantial. Steps should be taken to maintain the authenticity, accuracy, and accessibility of the log files. The investigators may even calculate the hash value of the evidence at the time of seizure and submit it along with the evidence, in order to prove its integrity.



Investigating IIS Logs: Best Practices

While handling IIS logs, the investigators must treat them carefully and consider these files as evidences

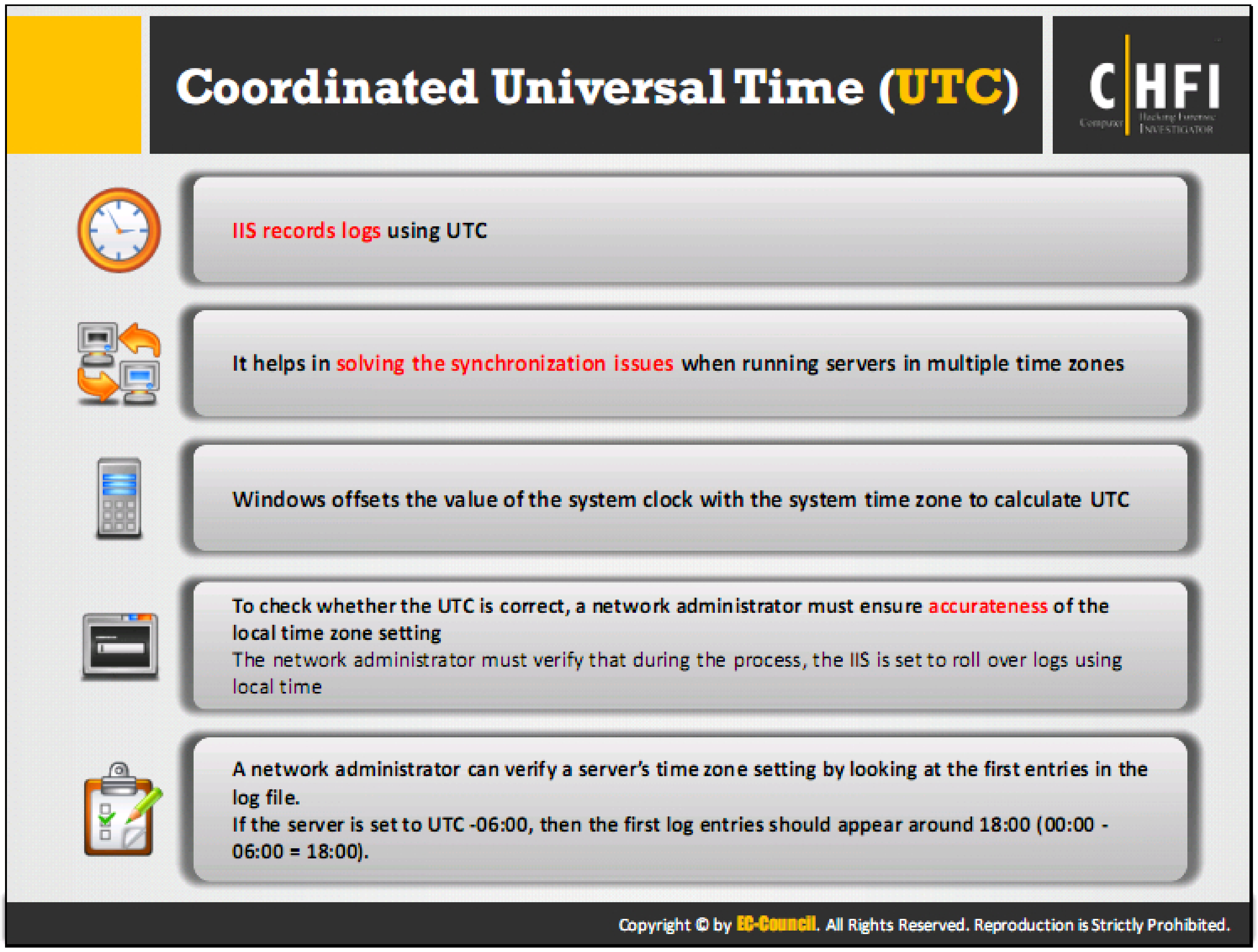
- IIS logs, in combination with other logs, such as firewall logs, IDS logs, and even TCPdump can provide more log credibility when used as an evidence
- Configure the IIS logs to record all the available fields
- Capture events with a accurate timestamp
- Maintain continuity in the logs
- Ensure IIS logs are not altered in any way from the time they have been originally recorded

Web server logs are huge in volume and examining such logs would be a tedious task. The slide contains some of the best practices for examining the logs.

In addition to the above discussed best practices, the forensic investigators can narrow down the logs search by following the steps mentioned below:

1. While investigating web attacks, a forensic examiner can go through the victim's incident report, so that he/she can narrow down the logs search.
2. Logs are generally stored in ASCII format, and each log file has column headers located at the top of that file. The investigators can write simple scripts to examine and parse the log files and filter the required information, such as source IP, status or response code, etc.
3. Use log viewers to view and examine logs
4. If investigators are aware of what they are searching for, they can use signatures to look for indications of specific activity.
5. When IIS records the logs in W3C Extended log file format, the IIS stores all the logged events in GMT format, instead of the local time zone format for the system.

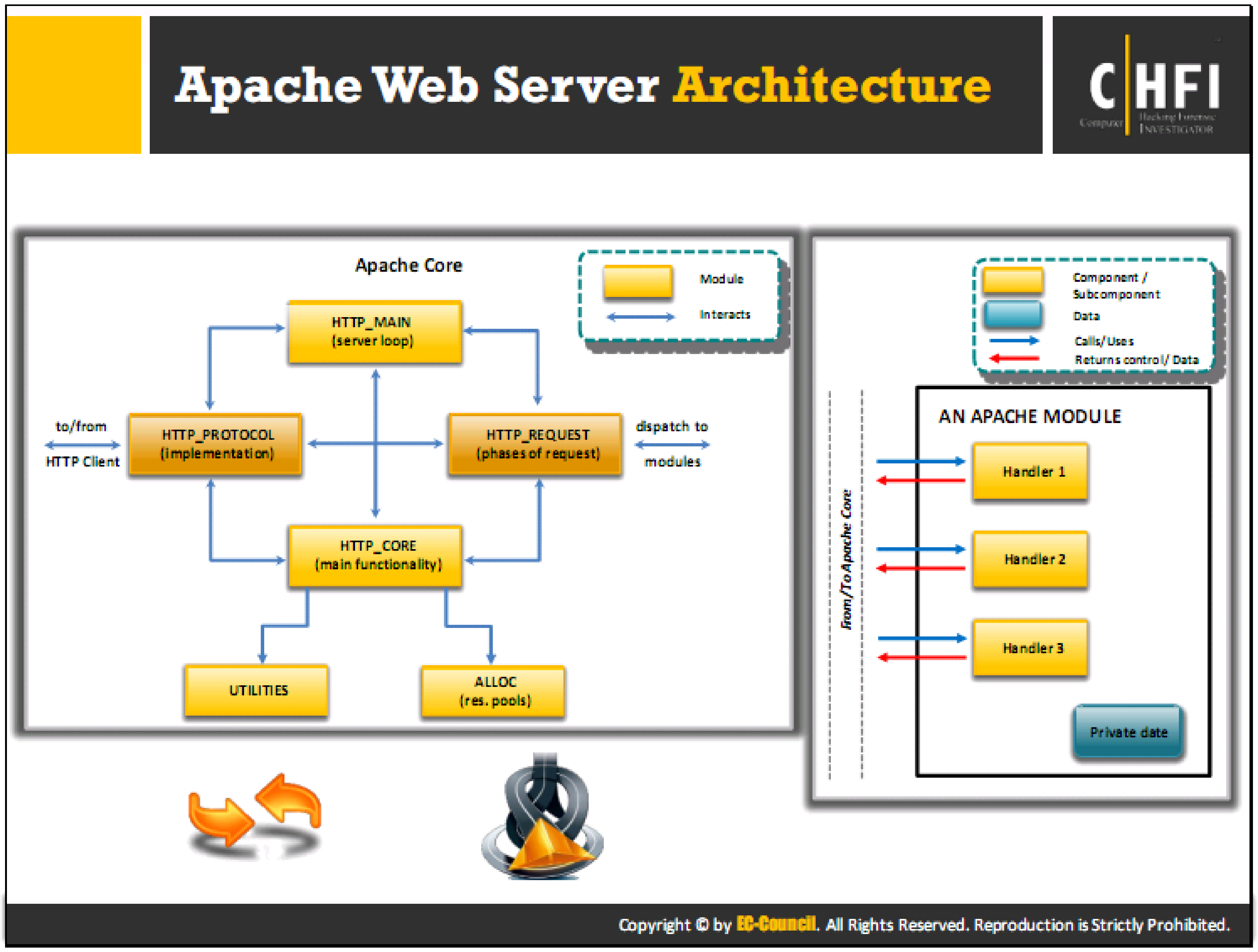
So, the investigators need to consider this point while examining the logs, since IIS creates a new log file on the next day at midnight GMT.



Coordinated Universal Time (UTC)

- IIS records logs using UTC
- It helps in solving the synchronization issues when running servers in multiple time zones
- Windows offsets the value of the system clock with the system time zone to calculate UTC
- To check whether the UTC is correct, a network administrator must ensure accurateness of the local time zone setting
  The network administrator must verify that during the process, the IIS is set to roll over logs using local time
- A network administrator can verify a server's time zone setting by looking at the first entries in the log file.
  If the server is set to UTC -06:00, then the first log entries should appear around 18:00 (00:00 - 06:00 = 18:00).

IIS records logs using UTC, which helps in synchronizing the servers in multiple zones. For calculating of UTC, the Windows offsets the value of the system clock with the system time zone. An accurate local time zone setting must be ensured by a network administrator, to validate the UTC. In addition, the administrator should also verify the process IIS is set to roll over logs using the local time. The server's time zone setting can be verified by looking at the first entries in the log file. If the server is set to UTC -06:00, then the first log entries should appear around 18:00 (00:00 - 06:00 = 18:00). Because UTC does not follow daylight savings, the administrator must also consider the date. For example, UTC -6:00 will be -5:00 half the year.



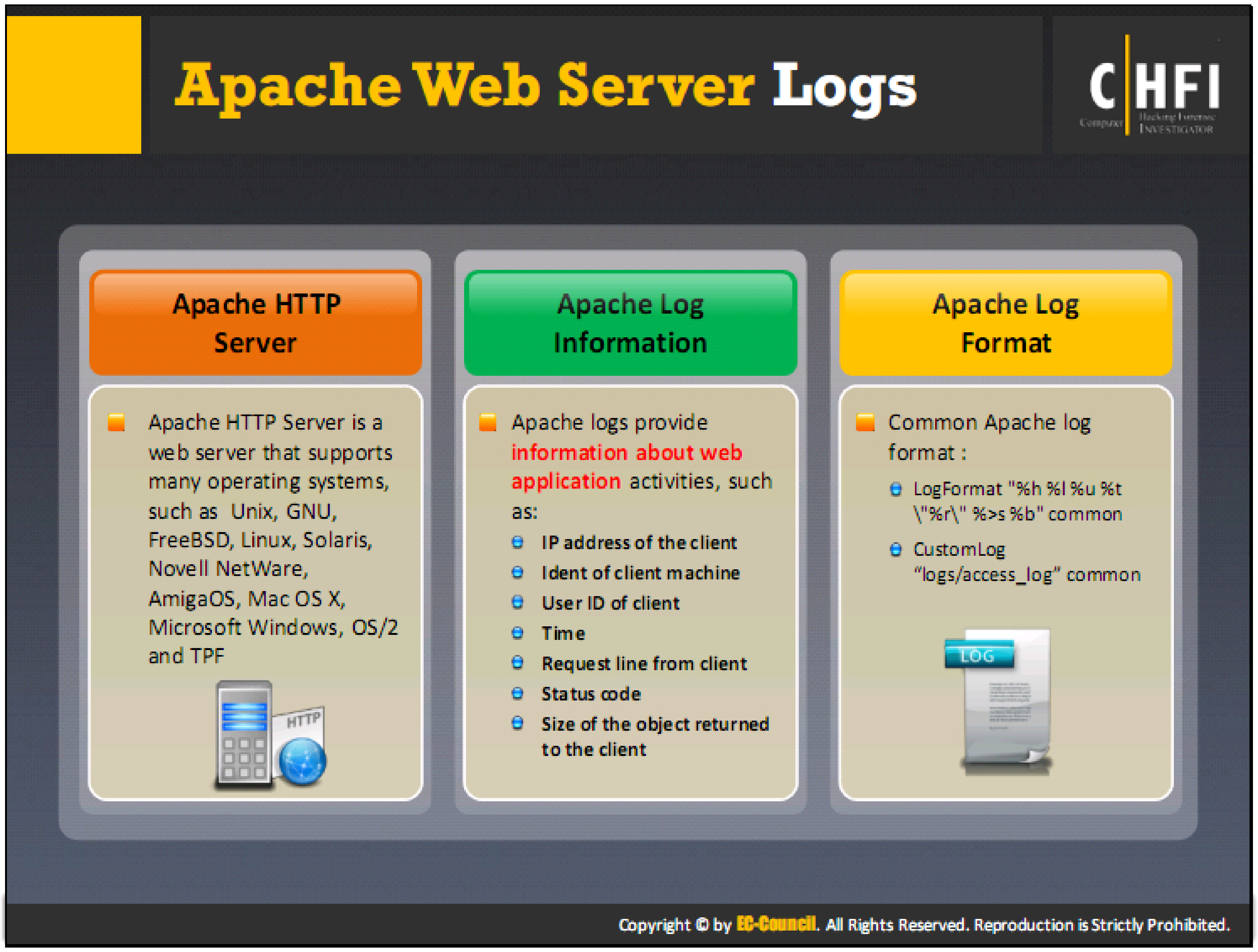
Apache Web Server **Architecture**

The Apache web server comprises of a modular approach. It consists of two major components, the Apache Core and the Apache Modules. The Apache Core deals with basic functionalities of the server, such as allocating the requests, maintaining and pooling the connections, etc., while the Apache Modules, which are simply add-ons (used for extending the core functionality of the server), looks after other functions, such as getting user ID from the HTTP request and validating the user, authorizing the user, etc. The Apache core consists of several components which have particular activities to perform. The elements of the Apache core are http_protocol, http_main, http_request, http_core, alloc, and http_config.

- **http_protocol:** This element is responsible for managing the routines, which interacts with the client and takes care of all the data exchange and socket connections between the client and the server.
- **http_main:** This element handles the server startups and timeouts. It also consists of the main server loop that waits for the connections and accepts them.
- **http_request:** This element controls the step by step procedure involved between the modules to complete a client request and is also responsible for error handling.
- **http_core:** This element is hardly functional enough to serve documents.
- **Alloc.c:** This element handles allocation of resource pools.

- **http_config:** This element is responsible for reading and handling of the configuration files. One of the main tasks of http_config is that it arranges all the modules, which the server will call during various phases of the request handling.

The second important component of the Apache server is the Apache Modules. As discussed, the architecture of the Apache web server has several modules that connect to the Apache core and assists the requests processed by the core. In order to change the Apache server's functionality, the developers may write new modules, which meet the desired purpose. According to the requirement of the request, the particular modules will be called. The modules implement the desired functionality and forward the output back to the core and the core assembles the output using the HTTP_REQUEST component of the Apache Core in order to send it to another module for processing or sends it back to the client. The modules are made up of handlers, which denote the particular functions to be performed by the module. The modules create specific handlers whenever a request is processed.



Apache Web Server Logs

Apache HTTP Server

- Apache HTTP Server is a web server that supports many operating systems, such as Unix, GNU, FreeBSD, Linux, Solaris, Novell NetWare, AmigaOS, Mac OS X, Microsoft Windows, OS/2 and TPF

Apache Log Information

- Apache logs provide information about web application activities, such as:
  - IP address of the client
  - Ident of client machine
  - User ID of client
  - Time
  - Request line from client
  - Status code
  - Size of the object returned to the client

Apache Log Format

- Common Apache log format :
  - LogFormat "%h %l %u %t \"%r\" %>s %b" common
  - CustomLog "logs/access_log" common

Apache HTTP Server

Apache HTTP Server is a web server that was originally developed for Unix/Linux operating systems; however, currently this can work under different operating systems like Mac, Windows, etc. It performs various functions (since it is a multi-threaded web server) requested by the client web browsers and can implement multiple tasks simultaneously. The Apache HTTP Server utilizes modules and extensions to support various environments.

Apache Log Information

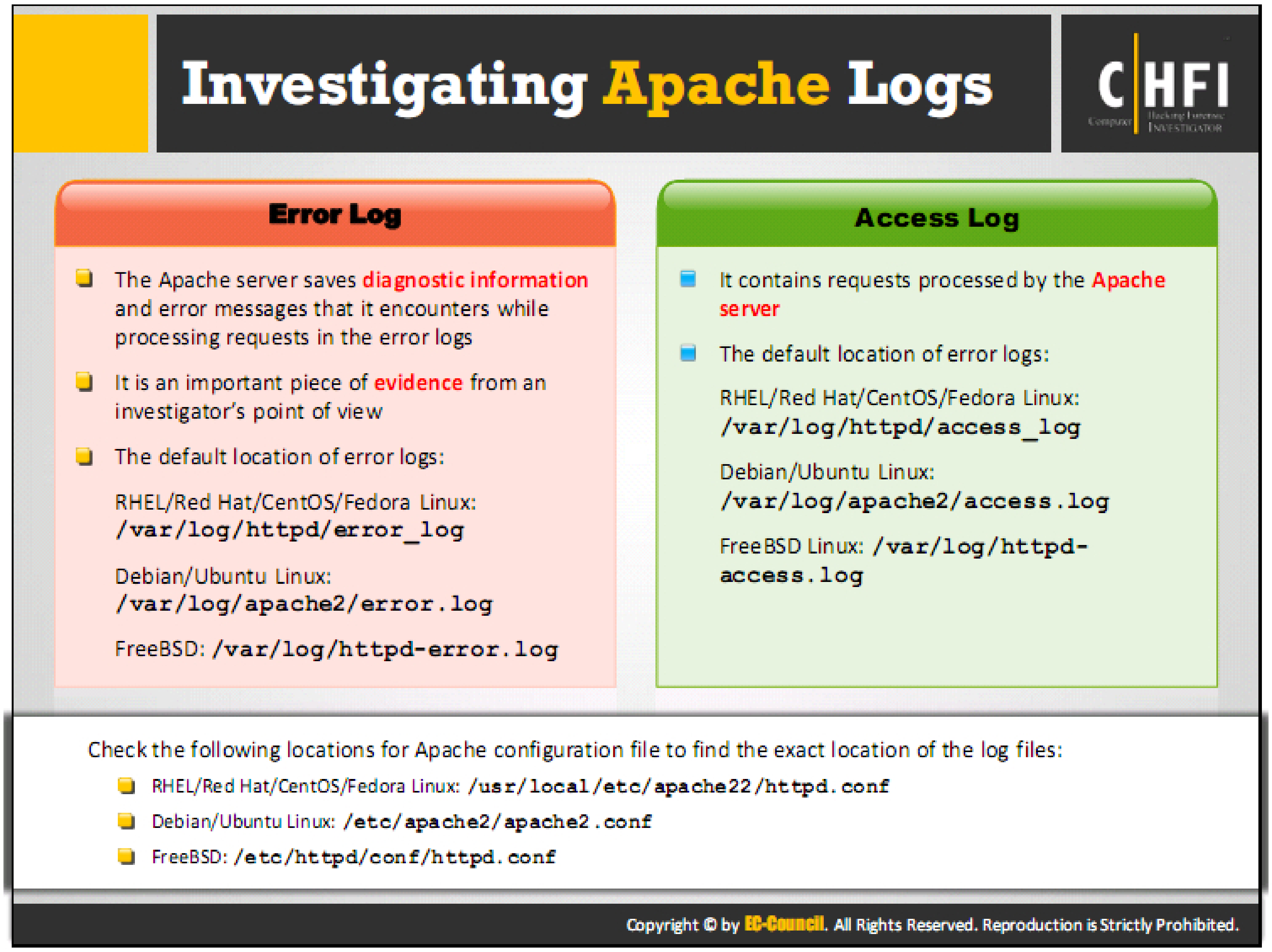
Apache logs provide very important information during auditing and forensic investigations about all the operations performed on the web server. This information includes client IP address, identd of the client machine, time, client user ID, Request line from a client, Status code, and Size of the object returned to the client. All the information provided by the logs can lead the investigator to the attacker.

Common Apache Log Format

%h %l %u %t \"%r\" %>s %b is the common percent directive log format. where:

- **%h** represents the client's IP address.
- **%l** represents the Remote log name. This will return a dash unless mod_ident is present and IdentityCheck is set on.
- **%u** is the client user ID.

- **%t** represents the time when the server received the request. It is displayed in the format [day/month/year:hour:minute:second zone].
- **\"%r\"** indicates the methods used for a request-response between a client and server, the resource requested by a client (apache_pb.gif), and the protocol used (HTTP/1.0).
- **%>s** represents the status code which the server sends back to the client.
- **%b** represents the size of the object which the server sends to the client.



Investigating Apache Logs

Error Log

- The Apache server saves diagnostic information and error messages that it encounters while processing requests in the error logs
- It is an important piece of evidence from an investigator's point of view
- The default location of error logs:

  RHEL/Red Hat/CentOS/Fedora Linux: `/var/log/httpd/error_log`

  Debian/Ubuntu Linux: `/var/log/apache2/error.log`

  FreeBSD: `/var/log/httpd-error.log`

Access Log

- It contains requests processed by the Apache server
- The default location of error logs:

  RHEL/Red Hat/CentOS/Fedora Linux: `/var/log/httpd/access_log`

  Debian/Ubuntu Linux: `/var/log/apache2/access.log`

  FreeBSD Linux: `/var/log/httpd-access.log`

Check the following locations for Apache configuration file to find the exact location of the log files:

- RHEL/Red Hat/CentOS/Fedora Linux: `/usr/local/etc/apache22/httpd.conf`
- Debian/Ubuntu Linux: `/etc/apache2/apache2.conf`
- FreeBSD: `/etc/httpd/conf/httpd.conf`

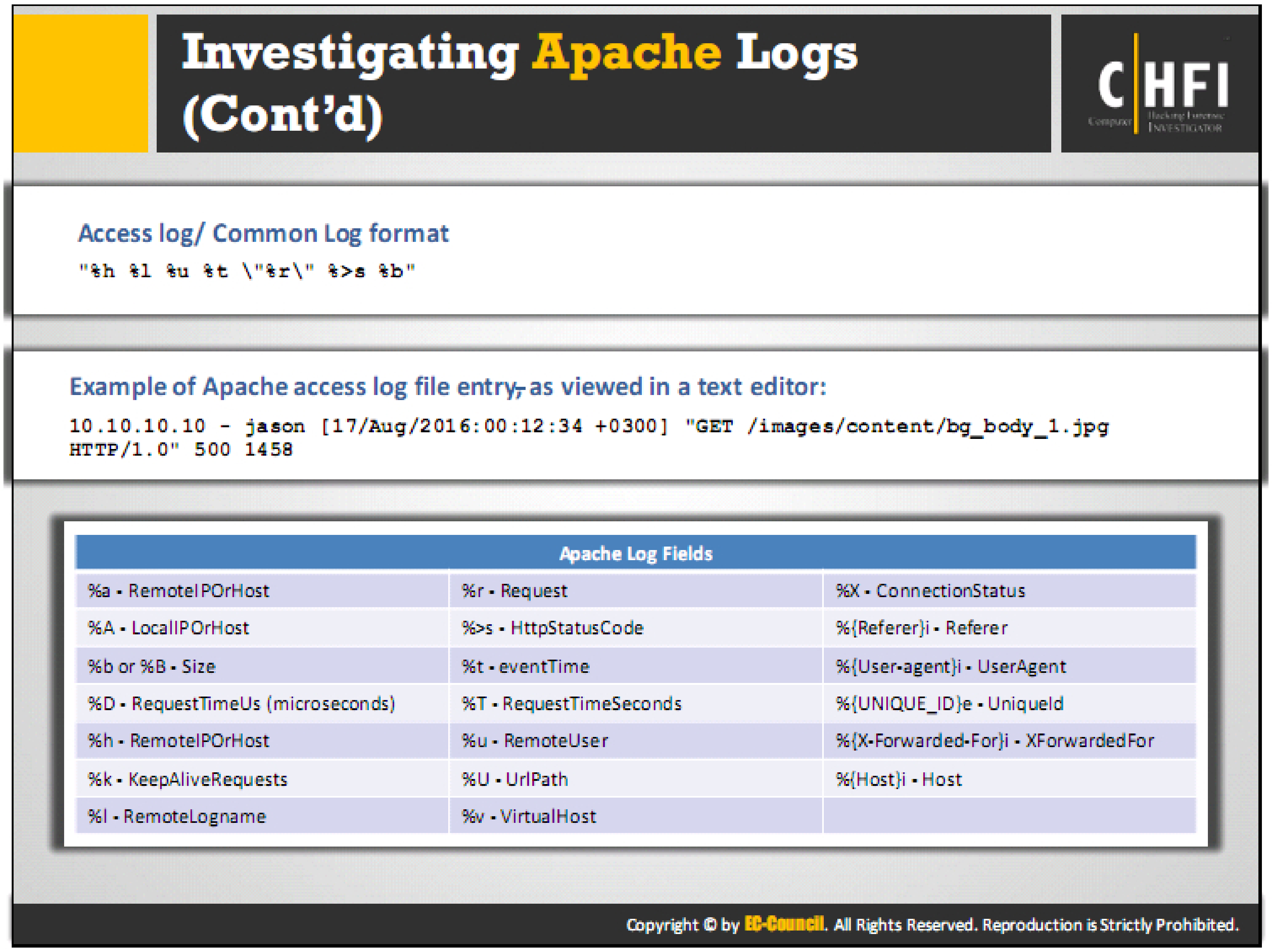
The Apache server consists of two logs:

Error Log

The Apache Error Log is the location where the server records all the errors that occurred during the client request processing. The ErrorLog directive sets the location of the error log. The log file contains data pertaining to the issues in the server's startup and operation. It also stores information related to the reason behind the issue and the steps involved in resolving it. The investigators need to use Linux applications like grep, cat, gedit or vi to read these log files.

Access Log

Requests processed by the Apache server are contained in the access log. It has a record of every request that goes through the server. The LogFormat directive helps in selecting the required log contents. The CustomLog directive sets the location and content of the Access log. The CustomLog directive also has the information to configure the server in such a way that the server can maintain access log records. The access logs are stored in the Common Log format by default and are highly configurable.



Investigating Apache Logs (Cont'd)

Access log/ Common Log format

```
"%h %l %u %t \"%r\" %>s %b"
```

Example of Apache access log file entry, as viewed in a text editor:

```
10.10.10.10 - jason [17/Aug/2016:00:12:34 +0300] "GET /images/content/bg_body_1.jpg
HTTP/1.0" 500 1458
```

Apache Log Fields		
%a ▪ RemoteIPOrHost	%r ▪ Request	%X ▪ ConnectionStatus
%A ▪ LocalIPOrHost	%>s ▪ HttpStatusCode	%{Referer}i ▪ Referer
%b or %B ▪ Size	%t ▪ eventTime	%{User-agent}i ▪ UserAgent
%D ▪ RequestTimeUs (microseconds)	%T ▪ RequestTimeSeconds	%{UNIQUE_ID}e ▪ UniqueId
%h ▪ RemoteIPOrHost	%u ▪ RemoteUser	%{X-Forwarded-For}i ▪ XForwardedFor
%k ▪ KeepAliveRequests	%U ▪ UrlPath	%{Host}i ▪ Host
%l ▪ RemoteLogname	%v ▪ VirtualHost	

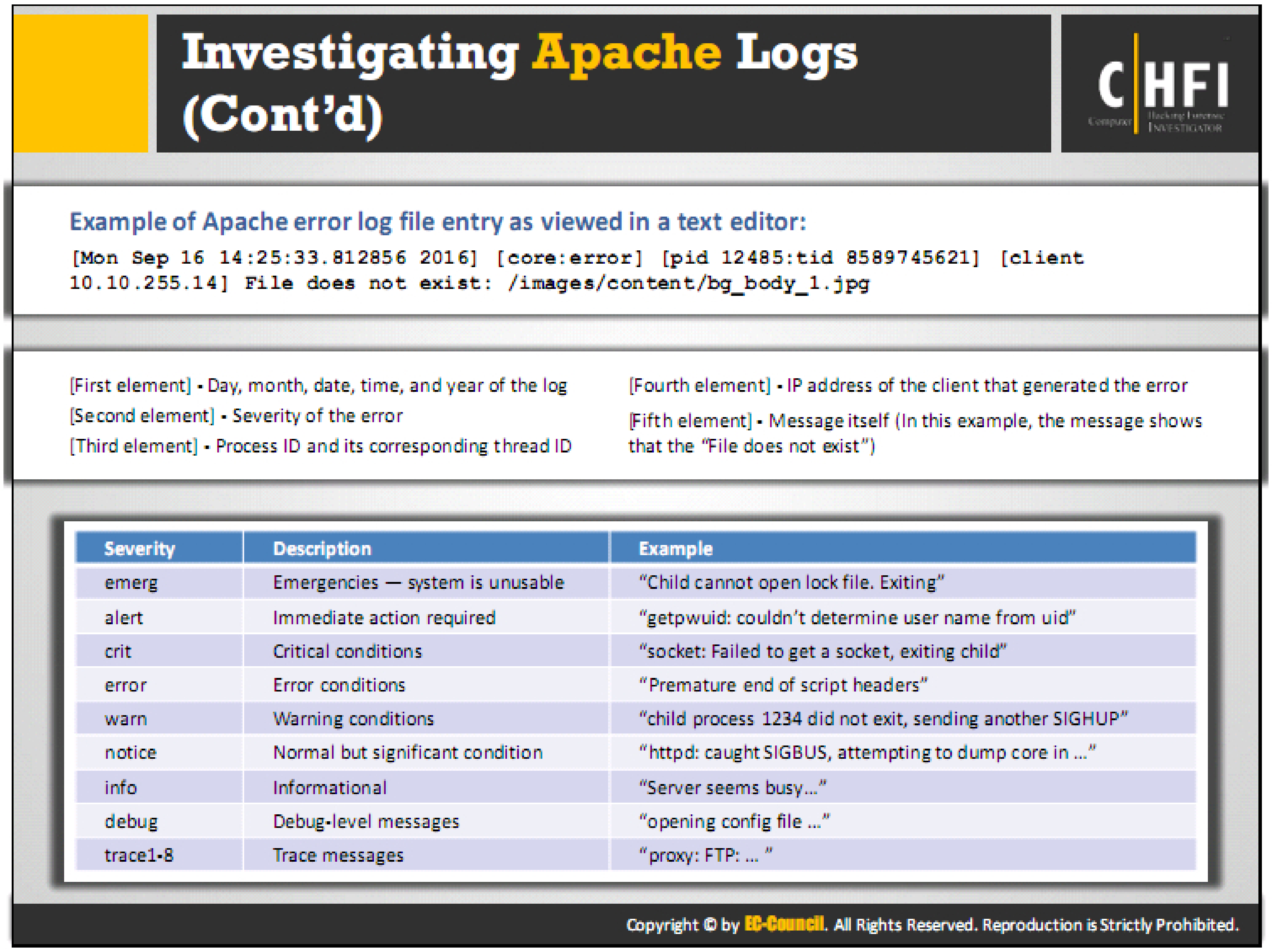
Apache Access Log/Common Log Format Example:

```
10.10.10.10 - jason [17/Aug/2016:00:12:34 +0300] "GET
/images/content/bg_body_1.jpg HTTP/1.0" 500 1458
```

A percent directive represents each field in the log. These percent directives enable the server to understand what information it has to log.

Let us map the percent directives with the actual log format.

- `10.10.10.10` (%h): IP Address of the client/remote host
- - (%l): The requested information is not available
- `jason` (%u): User ID of the person who sent the request
- `[17/Aug/2016:00:12:34 +0300]` (%t): The time at which the server finished processing the request. +03 UTC represents East Africa Time Zone.
- `"GET /images/content/bg_body_1.jpg HTTP/1.0"` (\"%r\"): The client used GET request method, and he/she requested the resource /images/content/bg_body_1.jpg. The client used HTTP/1.0 protocol.
- `200` (%>s): The status code represents that the response was successful
- `1458` (%b): The server returned the object of size 1458 bytes to the client



Investigating Apache Logs (Cont'd)

Example of Apache error log file entry as viewed in a text editor:

```
[Mon Sep 16 14:25:33.812856 2016] [core:error] [pid 12485:tid 8589745621] [client
10.10.255.14] File does not exist: /images/content/bg_body_1.jpg
```

[First element] - Day, month, date, time, and year of the log

[Second element] - Severity of the error

[Third element] - Process ID and its corresponding thread ID

[Fourth element] - IP address of the client that generated the error

[Fifth element] - Message itself (In this example, the message shows that the "File does not exist")

Severity	Description	Example
emerg	Emergencies — system is unusable	"Child cannot open lock file. Exiting"
alert	Immediate action required	"getpwuid: couldn't determine user name from uid"
crit	Critical conditions	"socket: Failed to get a socket, exiting child"
error	Error conditions	"Premature end of script headers"
warn	Warning conditions	"child process 1234 did not exit, sending another SIGHUP"
notice	Normal but significant condition	"httpd: caught SIGBUS, attempting to dump core in ..."
info	Informational	"Server seems busy..."
debug	Debug-level messages	"opening config file ..."
trace1-8	Trace messages	"proxy: FTP: ... "

Apache Error Log example:

```
[Mon Sep 16 14:25:33.812856 2016] [core:error] [pid 12485:tid 8589745621]
[client 10.10.255.14] File does not exist: /images/content/bg_body_1.jpg
```

Anatomy of the Log:

`Mon Sep 16 14:25:33.812856 2016`

This is the first element in the log entry. It contains the timestamp (day, month, date, time, and year) of the log.

`core:error`

The second element in the log describes the module producing the message. In this case, the Apache core is producing the message describing the security level (error).

`pid 12485:tid 8589745621`

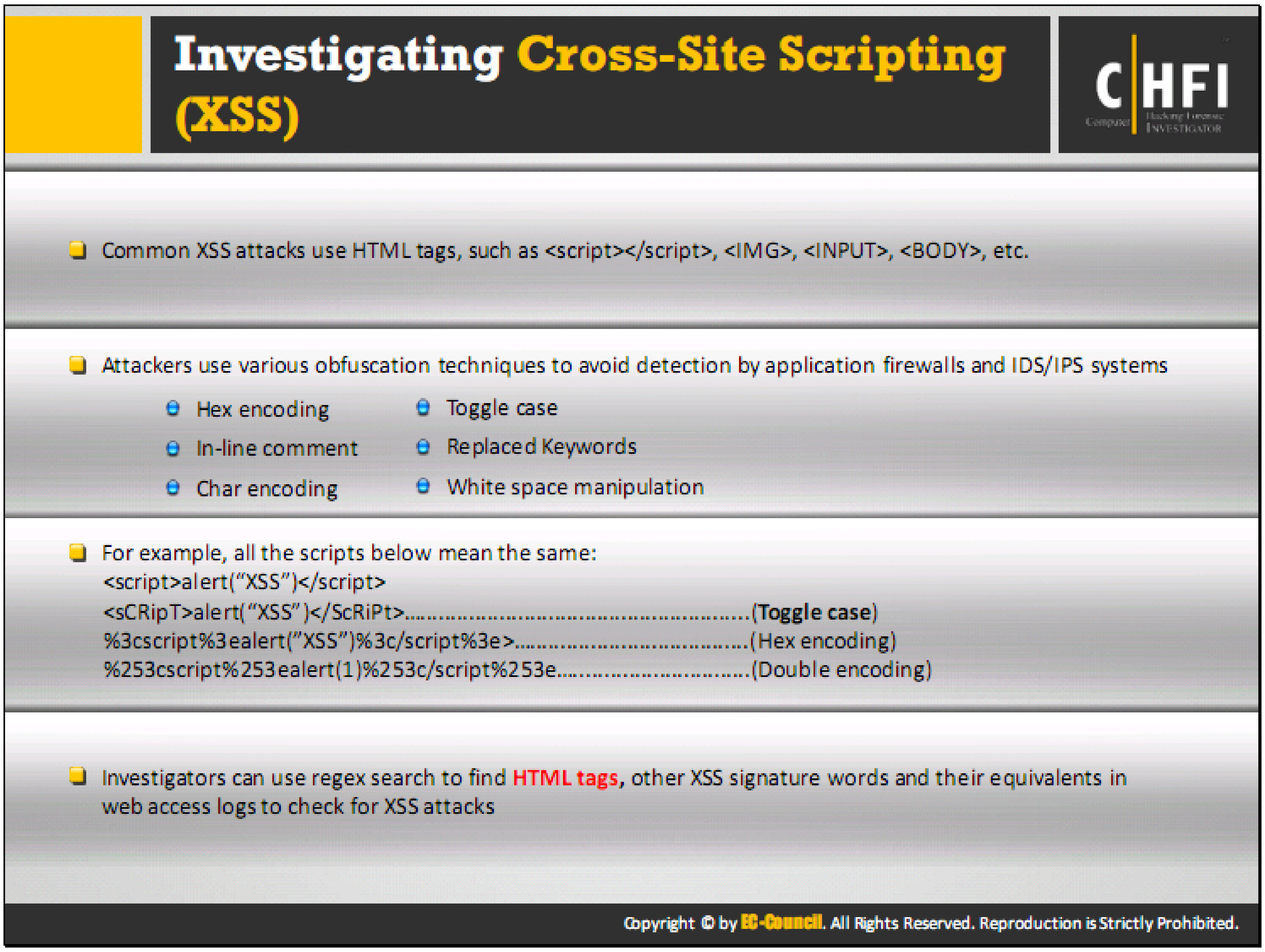
The next element in the log contains the process ID and its corresponding thread ID.

`client 10.10.255.14`

The fourth element in the log is the client address that made the request.

`File does not exist: /images/content/bg_body_1.jpg`

The final element in the log displays the status of the file, which the client has requested. In this case, the file does not exist. So, it displayed an error message stating the file does not exist on the server.



Investigating Cross-Site Scripting (XSS)

- Common XSS attacks use HTML tags, such as <script></script>, <IMG>, <INPUT>, <BODY>, etc.
- Attackers use various obfuscation techniques to avoid detection by application firewalls and IDS/IPS systems
  - Hex encoding
  - In-line comment
  - Char encoding
  - Toggle case
  - Replaced Keywords
  - White space manipulation
- For example, all the scripts below mean the same:

```
<script>alert("XSS")</script>
<sCRipT>alert("XSS")</ScRiPt>...................................................(Toggle case)
%3cscript%3ealert("XSS")%3c/script%3e>....................................(Hex encoding)
%253cscript%253ealert(1)%253c/script%253e.............................(Double encoding)
```

- Investigators can use regex search to find HTML tags, other XSS signature words and their equivalents in web access logs to check for XSS attacks

In XSS attack or Cross Site Scripting attack, the attacker exploits the vulnerability in the web by injecting malicious script, mostly Javascript, HTML or CSS markup in the web pages that is displayed in the user browser. This takes place when the user clicks on the malicious link.

With the implementation of proper firewalls, IDS, IPS, antivirus, etc., it becomes difficult for the attackers to perform attacks like XSS and SQL injection and bypass the security mechanisms. To avoid this, the attackers perform obfuscation techniques mentioned below to bypass them and perform malicious activities.

- **Hex Encoding**: Attackers use hex values of the characters to bypass the security mechanisms.

  Normal XSS script: `<script>alert("XSS")</script>`

  Hex encoded XSS script: `%3cscript%3ealert("XSS")%3c/script%3e>`

- **In-line comment**: Attackers use Inline comments in middle of attack strings to bypass security mechanisms.

  Code with inline comment:

  ```
  http://www.bank.com/accounts.php?id=/*!union*/+/*!select*/+1,2,concat(/
  *!table_name*/)+FrOm/*!information_schema*/.tables/*!WhErE*/+/*!TaBlE_s
  ChEMa*/+like+database()--
  ```

- **Char encoding/ double encoding**: Some of the Web Application Firewalls (WAFs) decode the hex encoded input and filters it, preventing an attack. To bypass them, the attackers might double encode the input. In such cases, some WAFs may not decode the input second time, inferring that the attackers successfully bypassed the WAF.

Code with char encoding:

```
http://www.bank.com/accounts.php?id=1%252f%252a*/union%252f%252a/select
%252f%252a*/1,2,3%252f%252a*/from%252f%252a*/users--
```

- **Toggle Case**: Some applications block the lowercase SQL keyword. In such case, attackers toggle the code to bypass them.

  Some firewalls contain the Regex Filter: /union\sselect/g. So, they may filter the suspicious code written in lower case letters.

  Code with toggle case:

  ```
  http://www.bank.com/accounts.php?id=1+UnIoN/**/SeLecT/**/1,2,3--
  ```

- **Replaced Keywords**: Some application and WAFs use preg_replace to remove all SQL keywords. Hence, the attackers use the following coding technique to bypass WAFs.

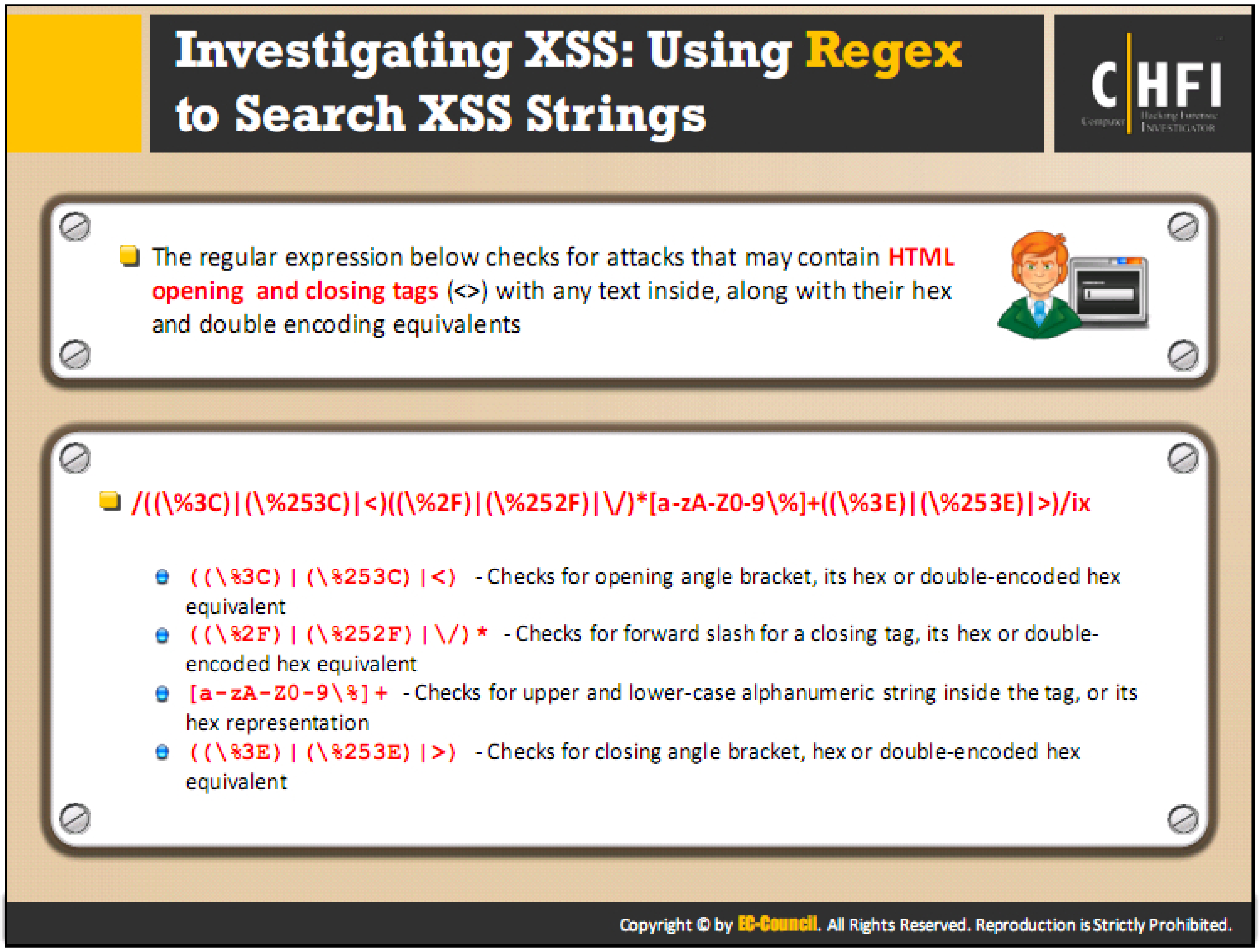
  Code with replaced Keywords:

  ```
  http://www.bank.com/accounts.php?id=1+UNunionION+SEselectLECT+1,2,3--
  ```

- **White space manipulation**: As explained above, when attackers replace the keywords, some WAFs may replace the keywords with white space. In such case, the attackers use "%0b" to eliminate the space and bypass the firewalls.

  Code with white space manipulation:

  ```
  http://www.bank.com/accounts.php?id=1+uni%0bon+se%0blect+1,2,3--
  ```



Investigating XSS: Using Regex to Search XSS Strings

- The regular expression below checks for attacks that may contain **HTML opening and closing tags** (<>) with any text inside, along with their hex and double encoding equivalents

- /((\%3C)|(\%253C)|<)((\%2F)|(\%252F)|\/)*[a-zA-Z0-9\%]+((\%3E)|(\%253E)|>)/ix
  - ((\%3C)|(\%253C)|<) - Checks for opening angle bracket, its hex or double-encoded hex equivalent
  - ((\%2F)|(\%252F)|\/)* - Checks for forward slash for a closing tag, its hex or double-encoded hex equivalent
  - [a-zA-Z0-9\%]+ - Checks for upper and lower-case alphanumeric string inside the tag, or its hex representation
  - ((\%3E)|(\%253E)|>) - Checks for closing angle bracket, hex or double-encoded hex equivalent

When an attacker attacks the dynamic web page, there is a chance that he may use HTML formatting tags, such as <b> for bold and <i> for italic. He may also use script tags, such as <script>alert("OK")</script>. Sometimes, he can use the hex equivalent of the code. For example, the hex equivalent of the <script> is %3C%73%63%72%69%70%74%3E.

Regular expression

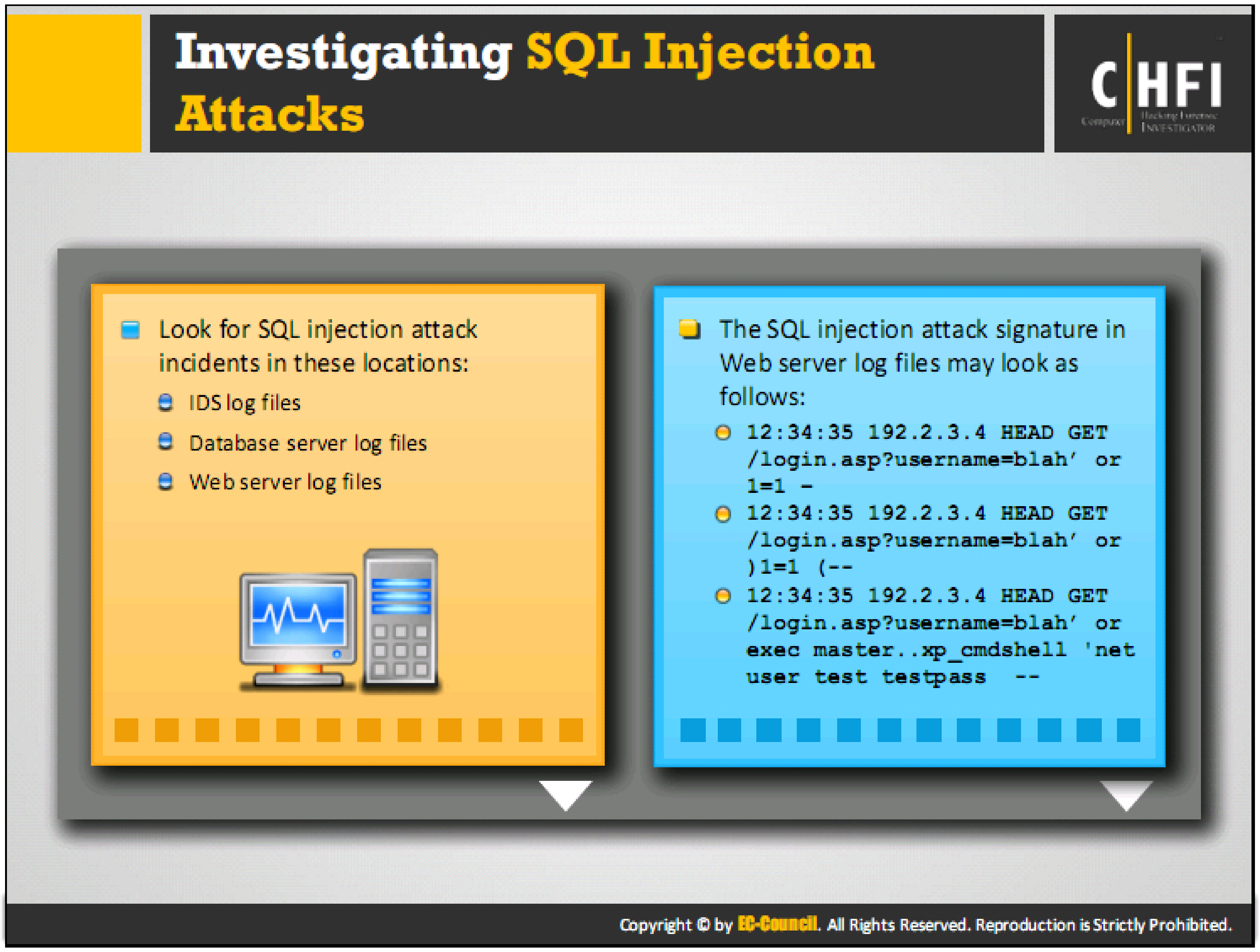
The following regular expression is the way to detect such types of attacks. It checks the HTML opening and closing tags <> containing the text inside so that it can easily catch the <b>, <i>, and <script> contents:

/((\%3C)|<)((\%2F)|\/)*[a-z0-9\%]+((\%3E)|>)/ix

Where:

Regex	Description
((\%3C)|<)	Checks for the opening **angle bracket** or its **hex equivalent**
((\%2F)|\/)*	Checks the forward slash for a closing tag or its hex equivalent
[a-z0-9\%]+	Checks for the string present inside the tag that may be an **alphanumeric string** or its **hex representation**
((\%3E)|>)	Checks for the closing angle bracket or its hex equivalent

TABLE 8.1: Regex Search Strings



Investigating SQL Injection Attacks

- Look for SQL injection attack incidents in these locations:
  - IDS log files
  - Database server log files
  - Web server log files

- The SQL injection attack signature in Web server log files may look as follows:

```
12:34:35 192.2.3.4 HEAD GET
/login.asp?username=blah’ or
1=1 -
12:34:35 192.2.3.4 HEAD GET
/login.asp?username=blah’ or
)1=1 (--
12:34:35 192.2.3.4 HEAD GET
/login.asp?username=blah’ or
exec master..xp_cmdshell 'net
user test testpass  --
```

The SQL injection attacks incidents can be found at three locations as mentioned below:

IDS log files

IDS logs permit the system administrators to identify any successful intrusions. The generated logs can help to identify the attack trends and patterns that assist in determining security holes where the perpetrators plan to attack the most. In addition, it also retrieves information related to any possible security holes or policy oversights, and any servers on the network that have a higher risk of being attacked.

Database server logs files

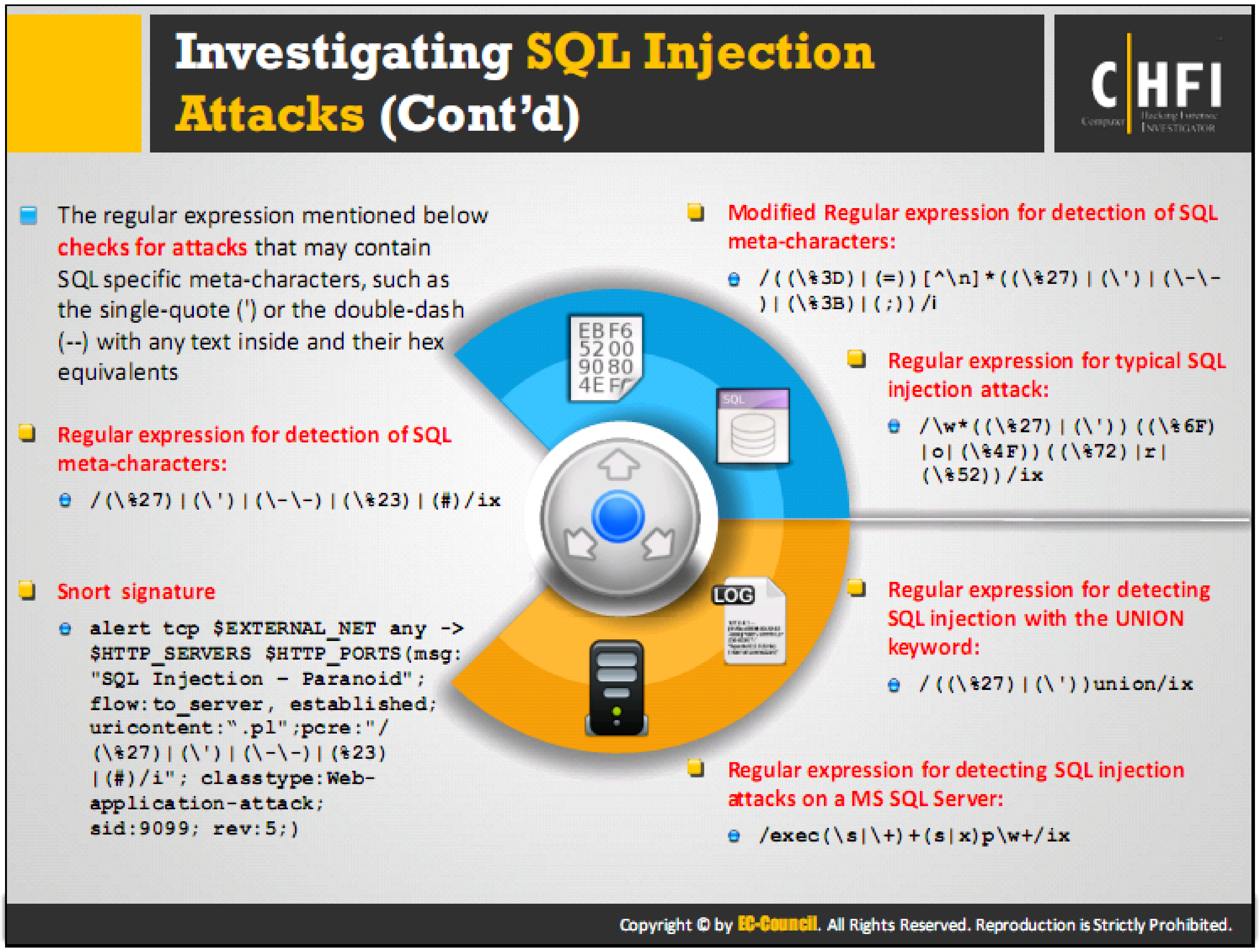
These log files record each message that is stored in the database and enables fault acceptance in case the database needs to be restored.

Web server logs files

Web server log help in understanding how and when the website pages and applications , along with other related information, such as which pages are being accessed, by whom and when. Each web server generates log files that keep a record of the information regarding access to a specific HTML page or graphic.

The attack signature may look like this in the log file:

12:34:35 192.2.3.4 HEAD GET /login.asp?username=blah‟ or 1=1 – 12:34:35 192.2.3.4 HEAD GET /login.asp?username=blah‟ or)1=1 (-- 12:34:35 192.2.3.4 HEAD GET /login.asp?username=blah‟ or exec master..xp_cmdshell 'net user test testpass --



Investigating SQL Injection Attacks (Cont'd)

C|HFI
Computer Hacking Forensic Investigator

- The regular expression mentioned below checks for attacks that may contain SQL specific meta-characters, such as the single-quote (') or the double-dash (--) with any text inside and their hex equivalents

- Regular expression for detection of SQL meta-characters:

```
/(\%27)|(\')|(\-\-)|(\%23)|(#)/ix
```

- Snort signature

```
alert tcp $EXTERNAL_NET any ->
$HTTP_SERVERS $HTTP_PORTS(msg:
"SQL Injection - Paranoid";
flow:to_server, established;
uricontent:".pl";pcre:"/
(\%27)|(\')|(\-\-)|(%23)
|(#)/i"; classtype:Web-
application-attack;
sid:9099; rev:5;)
```

- Modified Regular expression for detection of SQL meta-characters:

```
/((\%3D)|(=))[^\n]*((\%27)|(\')|(\-\-
)|(\%3B)|(;))/i
```

- Regular expression for typical SQL injection attack:

```
/\w*((\%27)|(\'))((\%6F)
|o|(\%4F))((\%72)|r|
(\%52))/ix
```

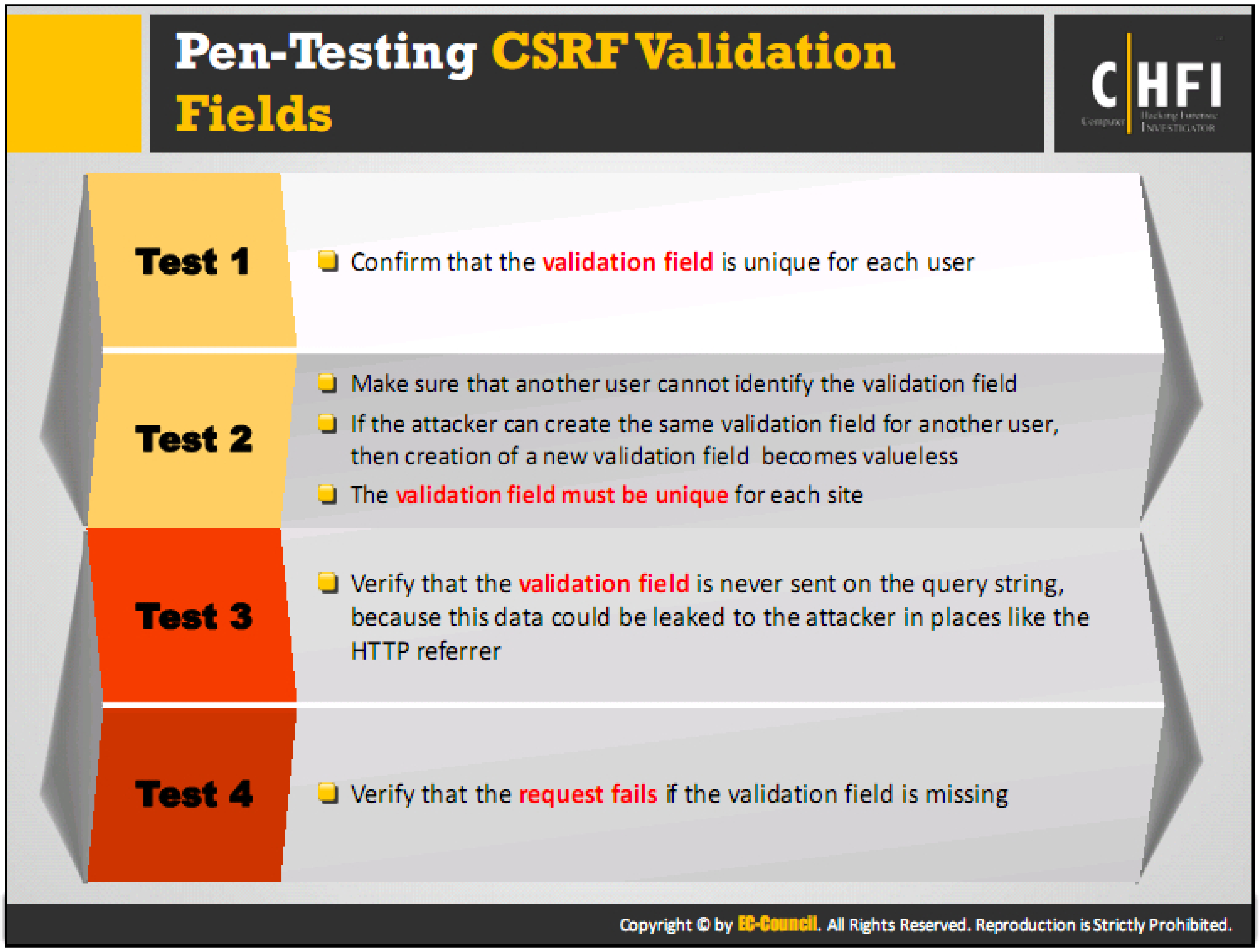
- Regular expression for detecting SQL injection with the UNION keyword:

```
/((\%27)|(\'))union/ix
```

- Regular expression for detecting SQL injection attacks on a MS SQL Server:

```
/exec(\s|\+)+(s|x)p\w+/ix
```

Use specific regular expressions to detect the SQL Injection attacks. The administrators need to write expressions in such a way that they can identify all kinds of meta-data used in an SQL query like a semi-colon, double –dash, single-quote, double minus sign, etc., as well as their hex equivalents. Use these regular expressions to frame snort signatures and rules to detect SQL injection attacks. Snort rules with different regular expressions are used to detect SQL injection attacks on the web application.



Pen-Testing CSRF Validation Fields

Test 1

- Confirm that the **validation field** is unique for each user

Test 2

- Make sure that another user cannot identify the validation field
- If the attacker can create the same validation field for another user, then creation of a new validation field becomes valueless
- The **validation field must be unique** for each site

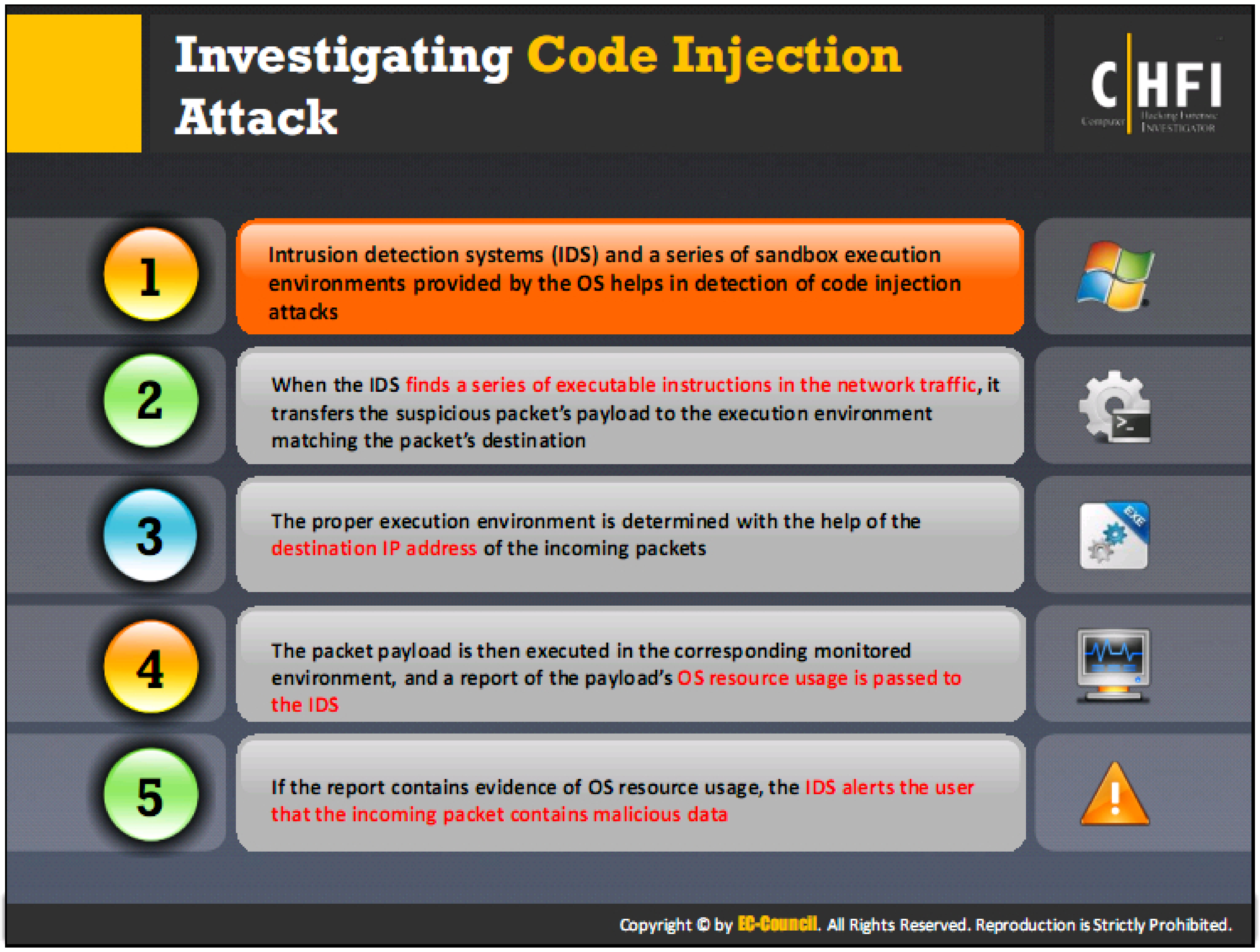
Test 3

- Verify that the **validation field** is never sent on the query string, because this data could be leaked to the attacker in places like the HTTP referrer

Test 4

- Verify that the **request fails** if the validation field is missing

Pen-Testing CSRF Validation Fields before filing the form is necessary to confirm the form validation before reaching the server. The above slide explains the process of Pen-testing of a form.

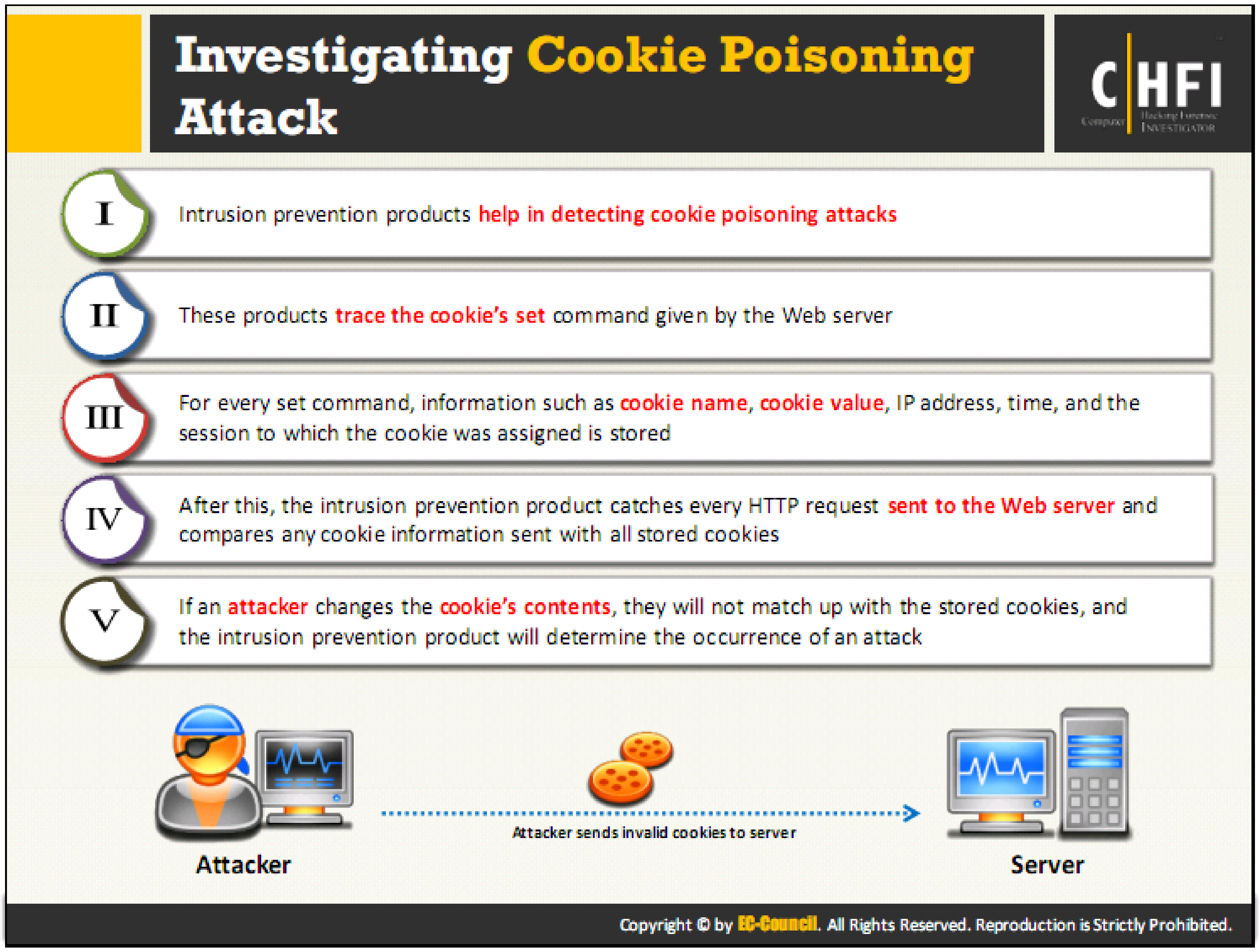


Investigating Code Injection Attack

1. Intrusion detection systems (IDS) and a series of sandbox execution environments provided by the OS helps in detection of code injection attacks
2. When the IDS finds a series of executable instructions in the network traffic, it transfers the suspicious packet's payload to the execution environment matching the packet's destination
3. The proper execution environment is determined with the help of the destination IP address of the incoming packets
4. The packet payload is then executed in the corresponding monitored environment, and a report of the payload's OS resource usage is passed to the IDS
5. If the report contains evidence of OS resource usage, the IDS alerts the user that the incoming packet contains malicious data

The executable instruction detector from the intrusion detection system and the series of sandbox execution environments that match the monitored application environments of the operating systems in the network are used for detecting the code injection attack. When the IDS find the series of executable instructions in the network traffic, it transfers those related packets' payloads to the execution environment matched with the packet's destination. The proper execution environment is determined with the help of the destination's IP address of incoming packets.

After that, the packet payload gets executed in the matching monitored environment, and the result of the payload's OS resource usage is passed to the IDS. If the report consists of evidence of the resource's usage, then the IDS alerts the user regarding the incoming packet that contains the malicious data. In this way, the investigator can detect the code injection attack that will lead to the perpetrator.



Investigating Cookie Poisoning Attack

I. Intrusion prevention products **help in detecting cookie poisoning attacks**

II. These products **trace the cookie's set** command given by the Web server

III. For every set command, information such as **cookie name**, **cookie value**, IP address, time, and the session to which the cookie was assigned is stored

IV. After this, the intrusion prevention product catches every HTTP request **sent to the Web server** and compares any cookie information sent with all stored cookies

V. If an **attacker** changes the **cookie's contents**, they will not match up with the stored cookies, and the intrusion prevention product will determine the occurrence of an attack

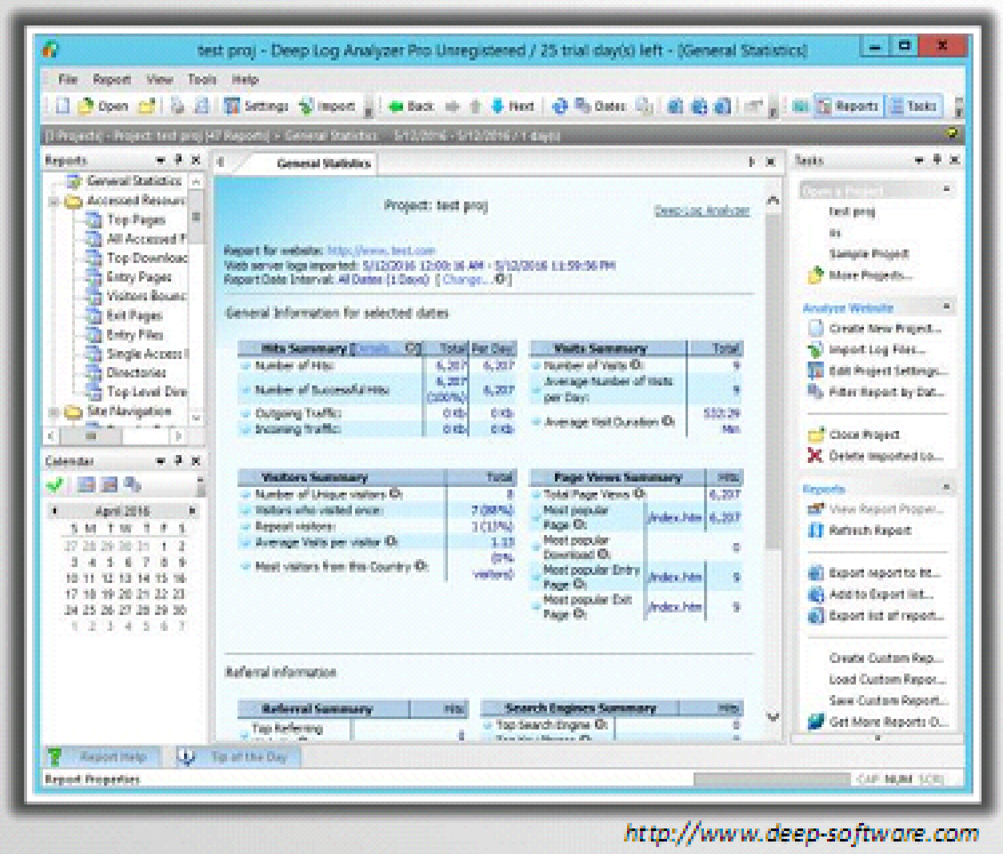
The detection of a cookie poisoning attack includes intrusion prevention products. These products trace the cookie's "set" command given by the web server. For every set command, these products store information such as cookie name, cookie value, IP address, and the session to which the cookie was assigned. It also stores the assigned time.

After this, the IPS catches every HTTP request sent to the web server, removes the information from it, and compares it with all the stored cookies. If the attacker changes the cookie's contents, then the IPS detects this changed information on a particular user and determines an attack has occurred. The investigator can view the IPS alerts regarding the Cookie Poisoning Attack to find the attacker.

Web Log Viewers

Deep Log Analyzer

It is a web analytics solution that enables you to analyze logs from web servers, such IIS on Windows, Apache or Nginx on Unix/Linux and more

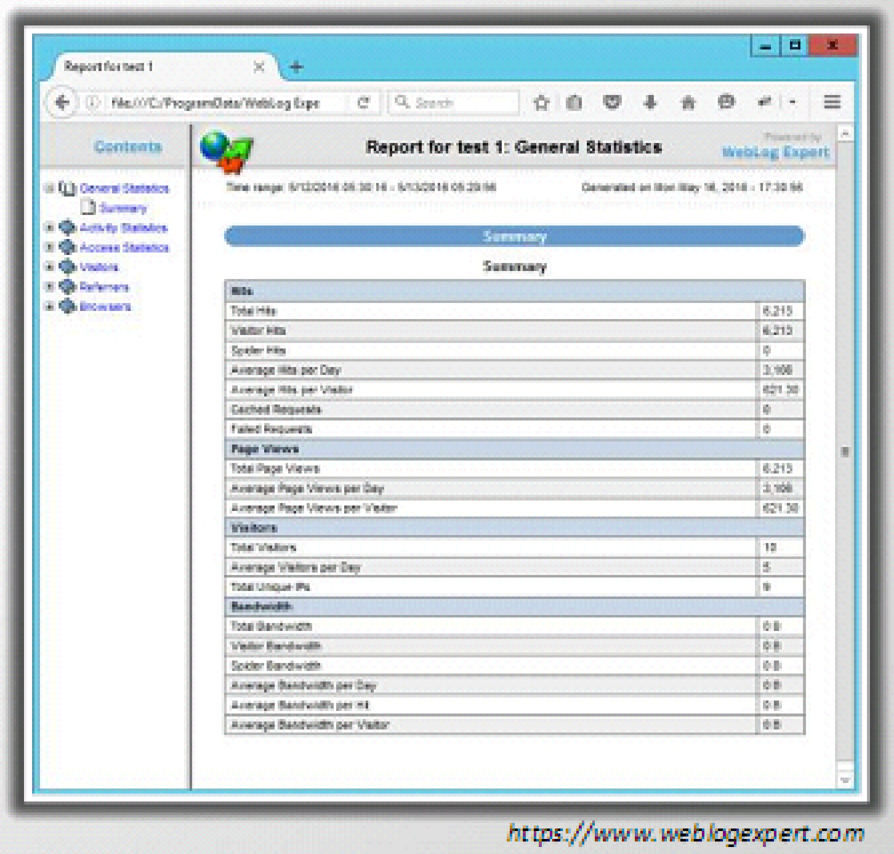


test proj - Deep Log Analyzer Pro Unregistered / 25 trial day(s) left - [General Statistics]

http://www.deep-software.com

WebLog Expert

It is an access log analyzer that enables you to analyze logs of Apache, IIS and Nginx web servers



Report for test 1: General Statistics

Summary

Hits	
Total Hits	6,215
Visitor Hits	6,215
Spider Hits	0
Average Hits per Day	3,108
Average Hits per Visitor	621.50
Cached Requests	0
Failed Requests	0
Page Views	
Total Page Views	6,215
Average Page Views per Day	3,108
Average Page Views per Visitor	621.50
Visitors	
Total Visitors	10
Average Visitors per Day	5
Total Unique IPs	9
Bandwidth	
Total Bandwidth	0 B
Visitor Bandwidth	0 B
Spider Bandwidth	0 B
Average Bandwidth per Day	0 B
Average Bandwidth per Hit	0 B
Average Bandwidth per Visitor	0 B

https://www.weblogexpert.com

Deep Log Analyzer

Source: http://www.apacheviewer.com

The Deep Log Analyzer is a web analytics solution for small and medium size websites. It analyzes web site visitors' behavior and gets the complete website usage statistics in easy steps.

Features:

- It provides website statistics and web analytics reports presentation with interactive navigation and hierarchical view
- It analyzes logs from popular web servers, such as IIS on Windows, Apache or Nginx on Unix/Linux, etc.
- It enables viewing of aggregated reports and allows its comparison reports for different intervals

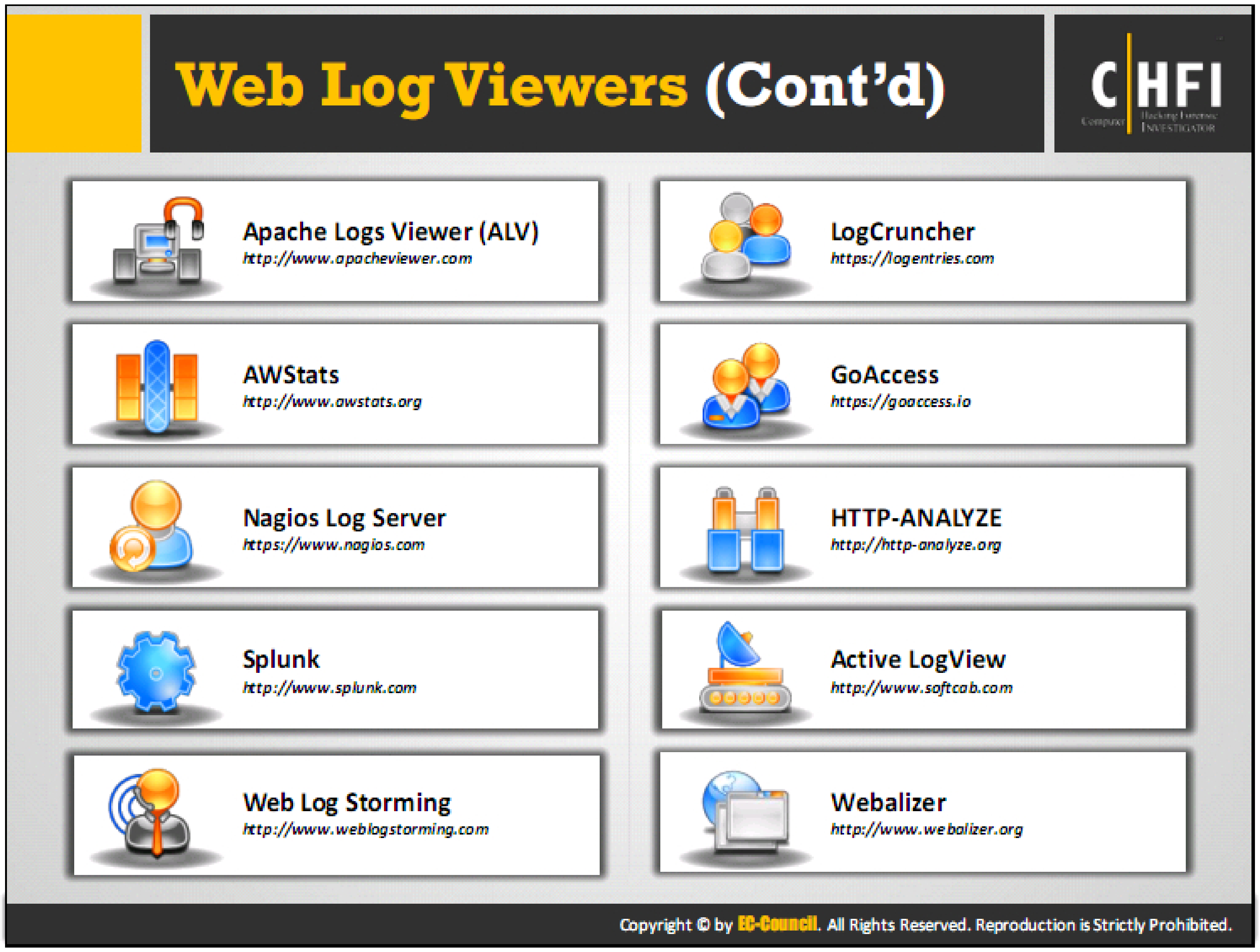
WebLog Expert

Source: https://www.weblogexpert.com

WebLog Expert is an access log analyzer which provides information about the site's visitors: activity statistics, accessed files, paths through the site, information about referring pages, search engines, browsers, operating systems, etc. The program generates reports that include both text information tables and charts.

Features:

- It provides general statistics, activity, and access statistics
- It gives information about visitors: hosts, top-level domains, countries, states, cities, authenticated users, screen resolutions, color depths and languages
- It gives information about errors
- It supports custom reports



Web Log Viewers (Cont'd)

Apache Logs Viewer (ALV)
http://www.apacheviewer.com

AWStats
http://www.awstats.org

Nagios Log Server
https://www.nagios.com

Splunk
http://www.splunk.com

Web Log Storming
http://www.weblogstorming.com

LogCruncher
https://logentries.com

GoAccess
https://goaccess.io

HTTP-ANALYZE
http://http-analyze.org

Active LogView
http://www.softcab.com

Webalizer
http://www.webalizer.org

Apache Logs Viewer (ALV)

Source: http://www.apacheviewer.com

Apache Logs Viewer (ALV) enables you to view, monitor, and analyze the Apache/IIS/nginx logs.

AWStats

Source: http://www.awstats.org

AWStats is a graphical tool that generates the web, streaming, ftp or mail server statistics. This log analyzer works as a CGI or from the command line and shows all possible information your log contains.

Nagios Log Server

Source: https://www.nagios.com

Nagios Log Server is a Centralized Log Management, Monitoring and Analysis Software. It simplifies the process of searching your log data. It sets up alerts to notify you when potential threats arise or simply query your log data to audit any system. Here, all log data are present in one location.

Splunk

Source: *http://www.splunk.com*

Splunk Enterprise helps in collection and analysis and acts upon the untapped value of the big data, which is generated by user's technology infrastructure, security systems, and business applications—giving you the insights to drive operational performance and business results.

Web Log Storming

Source: *http://www.weblogstorming.com*

Web Log Storming is a web server log file analyzer (IIS, Apache, and Nginx) for Windows.

LogCruncher

Source: *https://logentries.com*

LogCruncher is a tool for analysis and data visualization of web server log files. It allows the user to see and understand the website analytics based on key metrics.

GoAccess

Source: *https://goaccess.io*

GoAccess is an open source real-time web log analyzer and interactive viewer that runs in a terminal in *nix systems or through your browser. It provides HTTP statistics for system administrators that require a visual server report.

HTTP-ANALYZE

Source: *http://http-analyze.org*

The http-analyze is a logfile analyzer for web servers. It runs on any platform conforming to the ANSI C and POSIX standards ranging from personal computers to high-performance systems.

Active LogView

Source: *http://www.softcab.com*

Active LogView is a log analysis program that provides analysis of total requests, unique visits, advanced referrers list, hourly summary, user agents list, OS list, advanced filtration, advanced search and more.

Webalizer

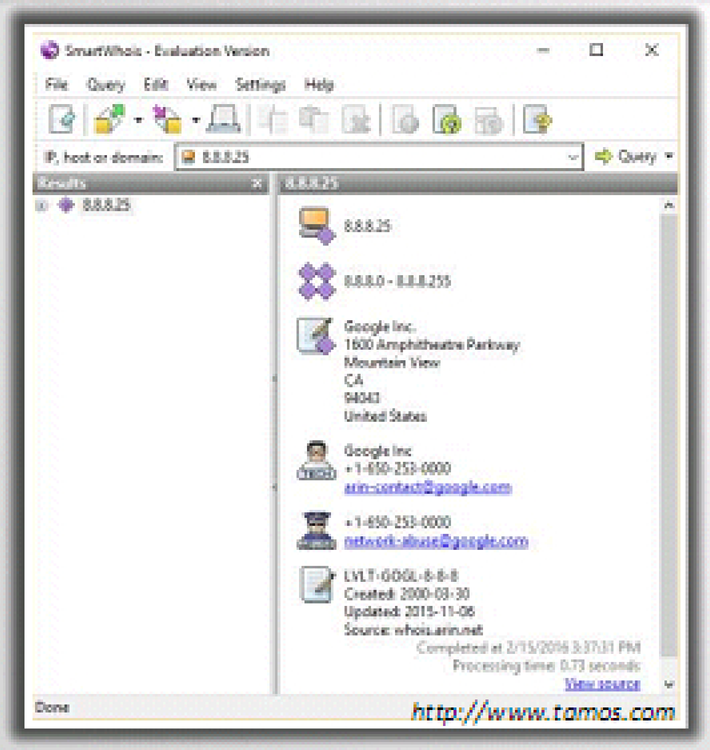
Source: *http://www.webalizer.org*

The Webalizer is a web server log file analysis program. It produces detailed, configurable usage reports in HTML format, for viewing with a standard web browser.

IP Address Locating Tools

SmartWhois

- Network information utility that allows to look up for all the available information about an IP address, hostname or domain, including country, state or province, city, name of the network provider, administrator and technical support contact information

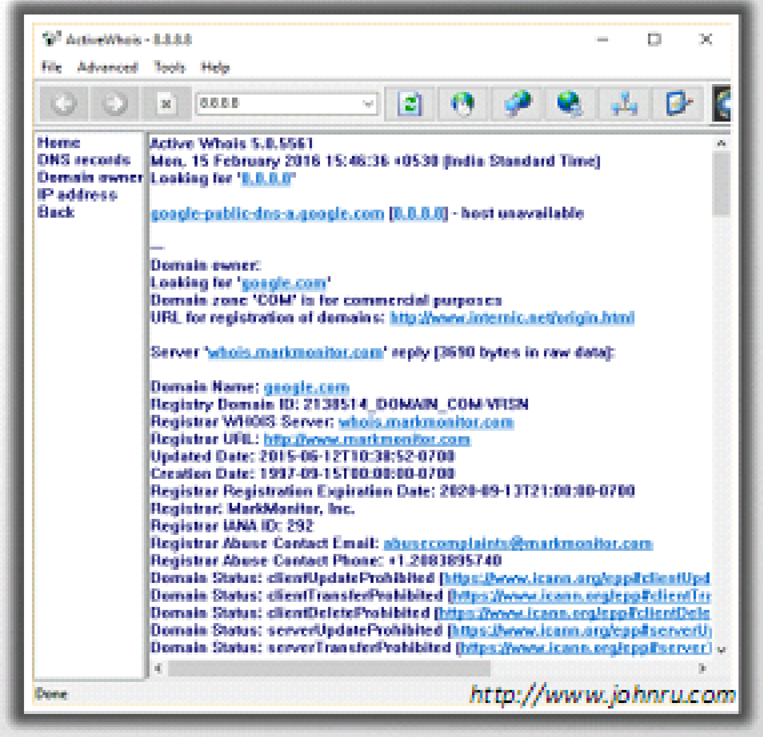


```
8.8.8.25

8.8.8.25

8.8.8.0 - 8.8.8.255

Google Inc.
1600 Amphitheatre Parkway
Mountain View
CA
94043
United States

Google Inc
+1-650-253-0000
arin-contact@google.com

+1-650-253-0000
network-abuse@google.com

LVLT-GOGL-8-8-8
Created: 2000-03-30
Updated: 2015-11-06
Source: whois.arin.net
                    Completed at 2/15/2016 3:37:31 PM
                         Processing time: 0.73 seconds
                                           View source
```

http://www.tamos.com

ActiveWhois

- Network tool to find any information about the owners of IP address or Internet domain
- You can determine the country, personal and postal addresses of owner, and/or user of IP address and domains



```
Active Whois 5.0.5561
Mon, 15 February 2016 15:46:36 +0530 (India Standard Time)
Looking for '8.8.8.8'

google-public-dns-a.google.com [8.8.8.8] - host unavailable

—
Domain owner:
Looking for 'google.com'
Domain zone 'COM' is for commercial purposes
URL for registration of domains: http://www.internic.net/origin.html

Server 'whois.markmonitor.com' reply [3690 bytes in raw data]:

Domain Name: google.com
Registry Domain ID: 2138514_DOMAIN_COM-VRSN
Registrar WHOIS Server: whois.markmonitor.com
Registrar URL: http://www.markmonitor.com
Updated Date: 2015-06-12T10:38:52-0700
Creation Date: 1997-09-15T00:00:00-0700
Registrar Registration Expiration Date: 2020-09-13T21:00:00-0700
Registrar: MarkMonitor, Inc.
Registrar IANA ID: 292
Registrar Abuse Contact Email: abusecomplaints@markmonitor.com
Registrar Abuse Contact Phone: +1.2083895740
Domain Status: clientUpdateProhibited (https://www.icann.org/epp#clientUpd
Domain Status: clientTransferProhibited (https://www.icann.org/epp#clientTr
Domain Status: clientDeleteProhibited (https://www.icann.org/epp#clientDele
Domain Status: serverUpdateProhibited (https://www.icann.org/epp#serverU)
Domain Status: serverTransferProhibited (https://www.icann.org/epp#server)
```

http://www.johnru.com

SmartWhois

Source: http://www.tamos.com

SmartWhois is a network information utility that allows you to look up all the available information about an IP address, hostname or domain, name of the network provider, administrator and technical support contact information. It supports Internationalized Domain Names (IDNs) and also fully supports IPv6 addresses.

Features:

- It saves results into an archive.
- Allows batch processing of IP addresses or domain lists.
- Enables caching of obtained results, hostname resolution, and DNS.
- It provides the customizable interface.

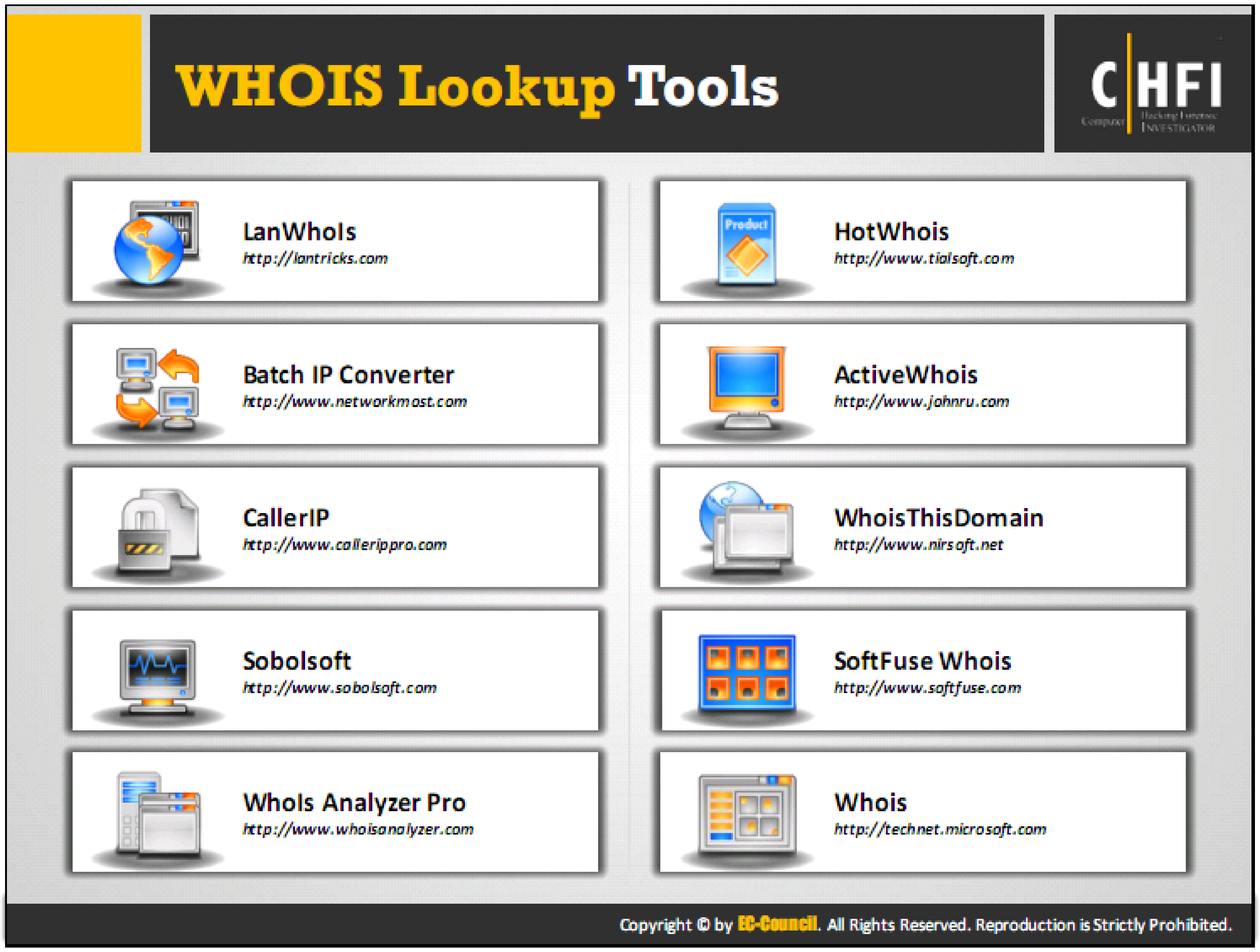
ActiveWhois

Source: http://www.johnru.com

ActiveWhois is a network tool for Windows which is used to find any information about the owners of IP address or Internet domain. You can determine the country, personal and postal addresses of the owner, and user of IP address and domains. ActiveWhois also allows users to explore DNS aliases.

Features:

- The WHOIS-hyperlink feature allows you to explore domain databases
- It allows to investigate even international domains.
- Active Whois provides direct links to the domain registrars for each country.
- ActiveWhois can also be used in offline mode.
- All the completed WHOIS requests will be saved to disk and can be instantly retrieved without the need for a live internet connection.
- The NetStat feature allows you to check who is connected to your computer.



WHOIS Lookup Tools

- LanWhoIs http://lantricks.com
- Batch IP Converter http://www.networkmost.com
- CallerIP http://www.callerippro.com
- Sobolsoft http://www.sobolsoft.com
- WhoIs Analyzer Pro http://www.whoisanalyzer.com
- HotWhois http://www.tialsoft.com
- ActiveWhois http://www.johnru.com
- WhoisThisDomain http://www.nirsoft.net
- SoftFuse Whois http://www.softfuse.com
- Whois http://technet.microsoft.com

LanWhoIs

Source: http://lantricks.com

The LanWhoIs program helps you find out who, where, and when registered the domain or site you are interested in, and the information about those who supports it currently.

Batch IP Converter

Source: http://www.networkmost.com

Batch IP Converter is a network tool to work with IP addresses. It combines Domain-to-IP Converter, Batch Ping, Tracert, Whois, Website Scanner and Connection Monitor into a single interface as well as an IP-to-Country Converter.

CallerIP

Source: http://www.callerippro.com

CallerIP informs you when someone has connected to your computer and can report the IP address. It also runs a trace on that IP address.

Sobolsoft

Source: http://www.sobolsoft.com

Sobolsoft is an online whois lookup tool.

WhoIs Analyzer Pro

Source: *http://www.whoisanalyzer.com*

WhoIs Analyzer Pro grants you access to contact records and other information from registrars and routing registries worldwide without you having to know which one to visit. It gives accurate information for any IP address, email address, URL, or Autonomous System Number (ASN) by giving you access to contact records from every country worldwide.

HotWhois

Source: *http://www.tialsoft.com*

HotWhois allows you to get all IP Whois and Domain whois information about IP addresses and domain names. This IP tracking tool can reveal valuable information, such as country, state, city, address, contact phone numbers and e-mail addresses of an IP provider.

ActiveWhois

Source: *http://www.johnru.com*

ActiveWhois is a network tool for Windows you can use to find any information about the owners of IP address or Internet domain. You can determine the country, personal and postal addresses of owner, and/or user of IP address and domains.

WhoisThisDomain

Source: *http://www.nirsoft.net*

WhoisThisDomain is a domain registration lookup utility allows you to easily get information about a registered domain. It automatically connects to the right WHOIS server, according to the top-level domain name, and retrieves the WHOIS record of the domain.

SoftFuse Whois

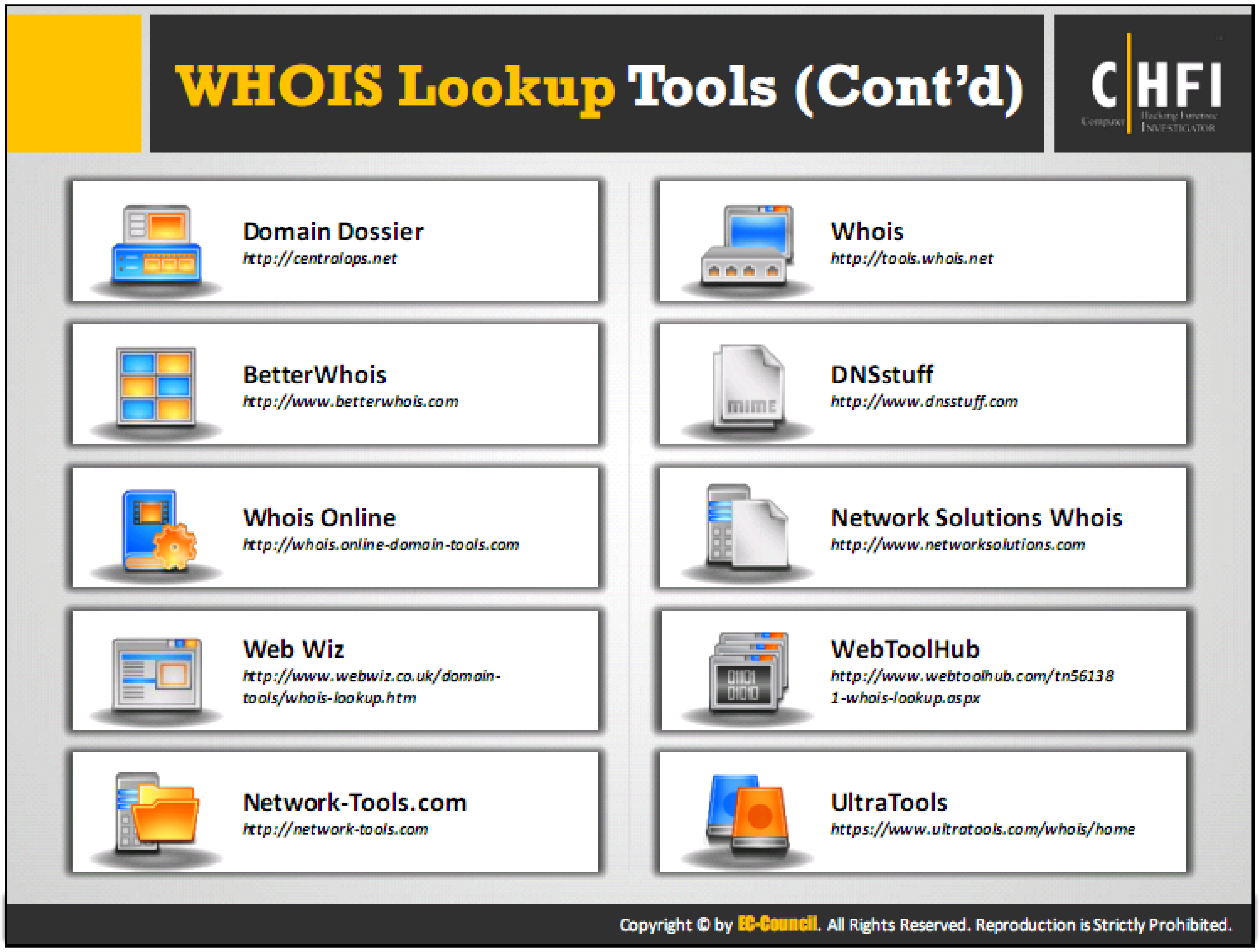
Source: *http://www.softfuse.com*

SoftFuse Whois is a desktop domain lookup utility. It does a lookup search for a domain and presents you with all available information, such as administrative, technical or billing contacts, domain location, hosting provider, creation, and expiration date.

Whois

Source: *http://technet.microsoft.com*

Whois performs the registration record for the domain name or IP address specified by you.



WHOIS Lookup Tools (Cont'd)

- **Domain Dossier**
  http://centralops.net
- **Whois**
  http://tools.whois.net
- **BetterWhois**
  http://www.betterwhois.com
- **DNSstuff**
  http://www.dnsstuff.com
- **Whois Online**
  http://whois.online-domain-tools.com
- **Network Solutions Whois**
  http://www.networksolutions.com
- **Web Wiz**
  http://www.webwiz.co.uk/domain-tools/whois-lookup.htm
- **WebToolHub**
  http://www.webtoolhub.com/tn561381-whois-lookup.aspx
- **Network-Tools.com**
  http://network-tools.com
- **UltraTools**
  https://www.ultratools.com/whois/home

Domain Dossier

Source: http://centralops.net

Domain Dossier is an online tool used to investigate domains and IP addresses.

BetterWhois

Source: http://www.betterwhois.com

BetterWhois offers a unified WHOIS search allowing you to check the domain availability, display domain ownership, and verify nameserver information across hundreds of domain registrars.

Whois Online

Source: http://whois.online-domain-tools.com

Whois Online is a tool that allows you to get information about various Internet resources, such as domain names, networks, IP addresses, domain registrants or autonomous systems. It queries WHOIS databases to get information that you are looking for. WHOIS record contains human-readable information about the organization (or person) that owns or administers the queried resource and the associated contact information.

Web Wiz

Source: http://www.webwiz.co.uk/domain-tools/whois-lookup.htm

WebWiz is an online tool used to look up the whois information for domains and IP addresses.

Network-Tools.com

Source: *http://network-tools.com*

Network-Tools.com is an online tool used to perform whois lookup on a target website.

Whois

Source: *http://tools.whois.net*

Whois performs the registration record for the user specified domain name or IP address.

DNSstuff

Source: *http://www.dnsstuff.com*

It allows forensic analysis of name and email servers, path analysis, authenticating and locating domains.

Network Solutions Whois

Source: *http://www.networksolutions.com*

Network Solutions Whois is an online tool used to look for domain availability.

WebToolHub

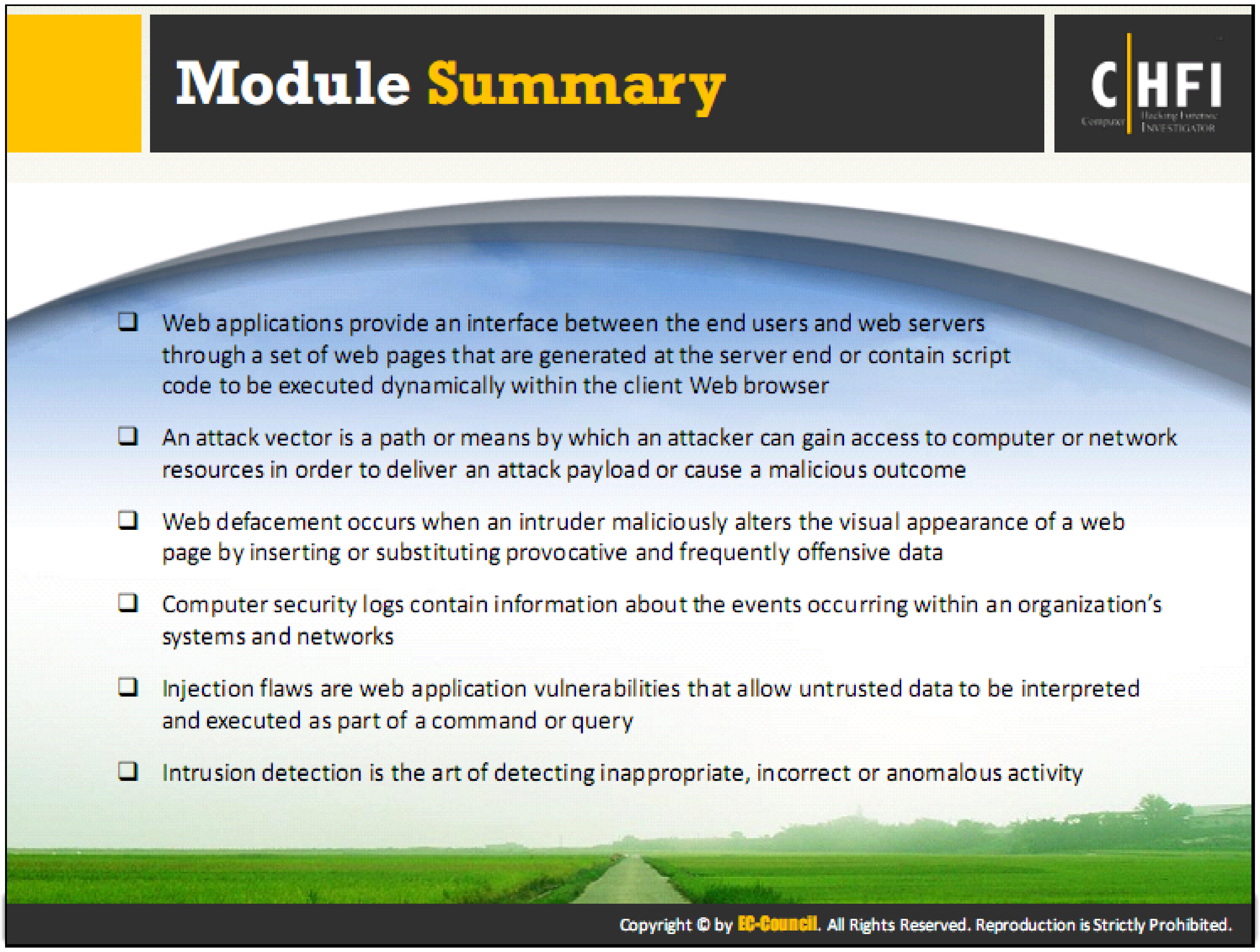
Source: *http://www.webtoolhub.com/tn561381-whois-lookup.aspx*

WebToolHub is an online whois lookup service to check the owner of the domain name or IP address.

UltraTools

Source: *https://www.ultratools.com/whois/home*

UltraTools is an online tool that shows you information about the domain you enter, including the Whois registration data, the Site Profile, and IP information.



Module Summary

- ❑ Web applications provide an interface between the end users and web servers through a set of web pages that are generated at the server end or contain script code to be executed dynamically within the client Web browser
- ❑ An attack vector is a path or means by which an attacker can gain access to computer or network resources in order to deliver an attack payload or cause a malicious outcome
- ❑ Web defacement occurs when an intruder maliciously alters the visual appearance of a web page by inserting or substituting provocative and frequently offensive data
- ❑ Computer security logs contain information about the events occurring within an organization's systems and networks
- ❑ Injection flaws are web application vulnerabilities that allow untrusted data to be interpreted and executed as part of a command or query
- ❑ Intrusion detection is the art of detecting inappropriate, incorrect or anomalous activity

In this module, we have learned about web applications and their architecture, web servers and its types, their working and reasons of compromise, different web attacks and ways to identify them, as well as web-based logs. This module will help us in understanding the role of web applications and their vulnerabilities in a security incident. This module also explains about various other methods that the attackers use to compromise the web application and web server security. Additionally, the module highlights various methods of gathering evidence from the web application and server-related incidents without disturbing the business.

The next module will discuss various databases currently in use, their architecture, targeted attacks and ways to investigate these attacks.