Supplementary Material of "*Detection of Attack-Targeted Scans from The Apache HTTP Server Access Logs*"

1. Basic Concepts

In this section, basic concepts which is necessary to fully understand the proposed model are introduced.

1.1. Apache HTTP Server Access Log

Logging information about HTTP requests could be beneficial for troubleshooting and optimizing a website. Web servers provide and store these important records through their access log files. Indeed, access logs give several details about the end users who had requested for the web sources in a web server. In this work, analysis of the access log files is based upon two most widely used web servers which are Apache HTTP Server and Internet Information Services (IIS) Server.

Changing the format of access logs enables to change their content. Common Log Format and Combined Log Format (CLF) are two common formats of Apache HTTP Server access logs [1]. Combined Log Format is similar to the other format with only two additional fields which are "Referer" and "User-agent". Throughout this work, because of its detailed content, Combined Log Format will be used.

The default log file format for IIS server is World Wide Web Consortium (W3C) Extended Log File Format [2]. Since W3C Extended Log File Format could be usually converted to CLF by some log conversion tools (e.g. Webalizer [3]), investigating only one of these web servers that is; Apache's CLF format, would be sufficient for the scope of this work.

The log file entries produced in CLF will look as follows:

«192.168.4.33 - - [30/Apr/2015:15:28:43 +0300] "GET /dvwa/ HTTP/1.0" 302 500 "-" "Opera/7.54 (Windows NT 5.1; U) [en]"»

Each part of this log entry is described below [1]:

- **192.168.4.33 (%h) »** The IP address of the client which made the request to the server.
- **- (%l) »** RFC 1413 identity of the client determined by The Identification Protocol (a.k.a., "ident") daemon on the clients machine. The hyphen "-" shows that the requested information is not available.
- **- (%u) »** The user id of the person in client side as determined by HTTP authentication.
- **[30/Apr/2015:15:28:43 +0300] (%t) »** The time when the request was made. The time is formatted as [day/month/year: hour: minute: second zone].
- **"GET /dvwa/ HTTP/1.0" (\"%r\") »** The request of the client is given in double quotes. GET is the HTTP method used. /dvwa/ is the requested resource from the server. HTTP/1.0 is the protocol used by the client.
- **302 (%>s) »** The status code sent by the server.
- **500 (%b) »** The size of the object returned to the client by the server. This size does not include the size of the response headers.
- **- (\"%{Referrer}i\") »** The information of the referrer that is the page providing a link to the requested page.
- **"Opera/7.54 (Windows NT 5.1; U) [en]" (\"%{User-agent}i\") »** The browser information provided by the client.

1.2. Hypertext Transfer Protocol

HTTP is an application layer protocol in both OSI Model and TCP/IP (Transmission Control Protocol/Internet Protocol) Hierarchy. HTTP is the protocol to exchange or transfer hypertext which is designed for the use of

hyperlinks [4]. It is a request-response protocol, stateless and object-oriented. A client requests for resources like Hyper-Text Markup Language (HTML) files, image files, etc. hosted in a web server and gets the related response via HTTP. HTTP has been in use by the World-Wide Web global information initiative for delivering data since 1990 [5]. The default port for HTTP is TCP/IP 80.

HTTP header fields contain information about the client browser, the requested page, the server and more. They are parts of HTTP headers found in both HTTP response and request. Thanks to HTTP header fields, forementioned access log entries are formed. The HTTP header fields used in the context of the work are listed below [6]:

Date: This general-header field shows the date and time of the request and the response. it must be formatted as defined in RFC 1123.

Host: This request-header field states the Internet host and port number of the requested resource and obtained from the original URI given by the client.

Referer: This request-header field allows specifying the address of the resource from which the Request-URI was obtained.

User-Agent: This request-header field contains information about the user agent originating the request.

1.3. HTTP Methods

HTTP methods identify the actions to be performed on the requested resource. Client's request contains this method token. If the server does not allow the requested method, it should give a response with the status code 405 (Method Not Allowed). According to the RFC 2616, there exist two types of HTTP methods; Safe Methods and Idempotent Methods [6]. Some of the important HTTP methods; that are also found in our access logs, are listed below [4]:

CONNECT: This method allows the connection to switch tunnelling (e.g. Secure Sockets Layer (SSL) tunnelling) with a proxy.

DELETE: This method asks the server for deleting the resource identified by the Request-URI.

GET: This method allows to retrieve the resource identified by the Request-URI.

HEAD: This method is similar to the GET method, but its response does not include the response body.

OPTIONS: This method allows the client to know the methods supported for a resource, or the capabilities of a server.

POST: This method asks the server for accepting the entity enclosed in the request as a new record of the Request-URI.

PRI: This method will appear to be used when an HTTP/1.1 server or intermediary attempts to parse an HTTP/2 connection preface [7]. This method cannot be used by an actual client.

PROPFIND: This method exists in Web Distributed Authoring and Versioning (WebDAV); that is an extension of the HTTP. It is used to retrieve properties, stored as Extensible Markup Language (XML), from a web resource [8].

PUT: This method asks the server for storing the entity enclosed in the request under the Request-URI.

TRACE: This method is used to invoke a remote, application layer loop-back of the request message. It allows the client for obtaining testing or diagnostic information about the server side.

TRACK: This method is identical to TRACE method, but it is used in IIS servers [9]

1.4. HTTP Status Codes

As a three-digit integer, status code represents how the server understands and satisfies the request. Its first digit defines the class of response. The other digits do not have any categorization role. Even if HTPP applications does not require to understand all of the status codes in RFC 2616, they must understand at least their classes. As stated below, there exist five different class of status codes[6]:

- **1xx (Informational):** The request was received, continuing process.

- **2xx (Successful):** The request was successfully received, understood, and accepted.
- **3xx (Redirection):** Further action needs to be taken in order to complete the request.
- **4xx (Client Error):** The request contains bad syntax or cannot be fulfilled.
- **5xx (Server Error):** The server failed to fulfil an apparently valid request.

1.5. Uniform Resource Identifiers (URI)

URIs are also known as World Wide Web (WWW) addresses, Universal Document Identifiers, and the combination of URLs and Uniform Resource Names (URN). URIs are formatted strings of characters to identify a web resource. Identification of a resource may be done by a name (URN), location (URL) or any other characteristics [6]. An usual form of URI is:

"scheme:[//[user:password@]host[:port]][/]path[?query]" [10].

According to RFC 2396 [11], URI has four components that are listed and shortly explained below:

1. **Scheme Component;** e.g. http, ftp, etc.,
2. **Authority Component;** a host name or an IP address followed by an optional port number,
3. **Path Component;** path that contains the target resource on the host side.

   "?": The first question mark is used as a separator between path and query strings, and is not part of the query string.
4. **Query Component;** a string of information to be interpreted by the resource. Two of the most common parameters in query strings are listed below:

   "&": A sequence of parameters and their assigned values are separated by the ampersand.

"=": The field name and value in query string are separated by an equals sign.

A locator, a name, or both could be used to classify a URI [11]. URL is used to identify resources via a representation of their primary access mechanism, while URN is used to serve as persistent, location-independent, resource identifiers [12].

1.6. Encoding

Encoding maps a scalar domain to a byte range (and vice versa). In fact, characters, such as letters, symbols and numbers, are converted to bytes according to the rule set defined with encoding. The reason for encoding is to make data transmission and storage more efficient. Data in the form of encoding used in web servers should be sent correctly to browsers and other user agents, so that they can interpret the bits and bytes properly. Even though, today's web servers and browsers support different encoding types, throughout this work, only UTF-8 (Unicode Transformation Format-8) and Hex Encodings will be taken into account.

1.6.1. UTF-8 Encoding

The Unicode Standard defines UTF-8. UTF-8 encodes UCS characters to the character in ISO/IEC 10646. Character numbers from U+0000 to U+007F (United States-American Standard Code for Information (US-ASCII) repertoire) correspond to octets 00 to 7F (7 bit US-ASCII values) [13].

1.6.2. Hex/Base 16 Encoding

Hex encoding uses 16-character subset of US-ASCII. A printable character is presented by four bits. The domain of the Base 16; that is 8-bit groups (octets), is mapped to the range of Base 16; that is strings of 2 encoded characters. From left to right, an 8-bit input is taken and is treated as 2 concatenated 4-bit groups, each of which is translated into a single character in the base 16 alphabet [14].

1.7. Robots Exclusion Standard

Robots Exclusion Standard (a.k.a. Robots Exclusion Protocol and robots.txt) represents an access policy for a website. Thanks to this standard, the way how web robots crawl or scan a website could be declared. Restrictions and permissions related to some resources or some robots could be announced via "robots.txt" file and it must be accessible via HTTP on the local URL "/robots.txt". An empty robots.txt file means that all web robots can visit all sources of the website. This file starts with one or more User-agent lines, followed by one or more Disallow lines, as detailed below [15]:

User-agent: This field is the subject of the following Disallow fields and it gets the name of a web robot as a value. if otherwise stated; when its value gets "*" sign, it means that all robots should pay attention to the following field.

Disallow: This field represents a resource restriction for the robots identified in the preceding User-agent fields. If this value is empty, all URLs can be retrieved by the specified web robots. According to the standard [15], at least one Disallow field needs to be present in robots.txt file.

2. Data and Log Generation

In this section, tools, applications, virtual environment used throughout this work and their installation and configuration settings are explained.

2.1. Virtual Machines

In order to run virtual machines, VMware Workstation 12 Player (Version 12.1.0 build-3272444) has been chosen. The reason for this choice is that VMware virtualization application offers improved virtual machine performance with an easy set-up. VMware has been running on a computer in which Windows 10 Pro was installed. In the context of this work, two virtual machines were created. One of them is used as a web server, the other is generally used for vulnerability scanning purposes.

2.2. Guest Operating Systems

One of the guest operating systems is Ubuntu 14.04 LTS whose OS type is 64-bit since it has new versions of many applications, newer kernel and also long term support. This virtual machine has 2 GB memory, and 20 GB hard disk and its network adapter setting is configured as bridged.

The other guest operating system is Kali GNU/Linux 2.0 to take advantage of its newer Debian packages and more advanced penetration testing platform. This virtual machine has 1.9 GB memory, and 20 GB hard disk and its network adapter setting is configured as bridged too.

References

[1] The Apache Software Foundation, Log Files, accessed December 15, 2015.
URL `https://httpd.apache.org/docs/2.4/en/logs.html`

[2] Microsoft Corporation, W3C Extended Log File Format (IIS 6.0), accessed December 10, 2016.
URL `https://www.microsoft.com/technet/prodtechnol/WindowsServer2003/Library/IIS/a3ca6f3a-7fc3-4514-9b61-f586d41bd483.mspx?mfr=true`

[3] Wikipedia, Webalizer, accessed January 21, 2016 (2016).
URL `https://en.wikipedia.org/wiki/Webalizer`

[4] Wikipedia, Hypertext Transfer Protocol, accessed December 11, 2016 (2016).
URL `https://en.wikipedia.org/wiki/Hypertext_Transfer_Protocol`

[5] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, T. Berners-Lee, Hypertext Transfer Protocol – HTTP/1.1, accessed December 15, 2015 (1997).
URL `https://tools.ietf.org/html/rfc2068`

[6] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, T. Berners-Lee, Hypertext Transfer Protocol – HTTP/1.1, accessed De-

cember 15, 2015 (1999).
URL https://www.w3.org/Protocols/rfc2616/rfc2616.html

[7] M. Belshe, R. Peon, E. M. Thomson, Hypertext Transfer Protocol Version 2 (HTTP/2), accessed December 11, 2016 (2015).
URL https://tools.ietf.org/html/rfc7540#section-11.6

[8] Wikipedia, WebDAV, accessed December 11, 2016 (2016).
URL https://en.wikipedia.org/wiki/WebDAV

[9] The Open Web Application Security Project (OWASP), Cross Site Tracing, accessed December 11, 2016 (2014).
URL https://www.owasp.org/index.php/Cross_Site_Tracing

[10] Wikipedia, Uniform Resource Identifier, accessed December 12, 2016 (2016).
URL https://en.wikipedia.org/wiki/Uniform_Resource_Identifier

[11] T. Berners-Lee, R. Fielding, L. Masinter, Uniform Resource Identifiers (URI): Generic Syntax, accessed December 12, 2016 (1998).
URL https://www.ietf.org/rfc/rfc2396.txt

[12] R. Moats, URN Syntax, accessed December 27, 2016 (1997).
URL https://www.ietf.org/rfc/rfc2141.txt

[13] F. Yergeau, UTF-8, a transformation format of ISO 10646, accessed January 21, 2016 (2003).
URL https://tools.ietf.org/html/rfc3629

[14] S. Josefsson, The Base16, Base32, and Base64 Data Encodings, accessed January 29, 2016 (2006).
URL https://tools.ietf.org/html/rfc4648

[15] Wikipedia, Robots exclusion standard, accessed February 17, 2016 (2016).
URL https://en.wikipedia.org/wiki/Robots_exclusion_standard