Web Attack Forensics

MODULE 15

Contents

Web Attack Forensics

15.1 LEARNING OBJECTIVES

After going through this unit, you will be able to:

- Define web services
- Explain various attacks on web services
- Define major tasks while performing web application forensics
- Explain application forensics readiness
- Explain methodologies for investigation of web application
- Perform website traffic analysis, and
- Evaluate various web application forensics tools

15.2 WEB ATTACK FORENSICS

Although there are mechanisms to protect our applications etc. from web attacks but it’s quite difficult to find the attacker and book him/her under law. The difficulty in traceability of the hackers/offenders prompts them to do more crimes. The major objective of web forensics is to trace the attacker and in line collect enough evidence that can be presented and accepted in the court of law. The aspects of investigation into web attacks can be viewed in two areas; a) web application forensics and web services forensics.

Although web forensics is a vital necessity the trends are complex and very vast. The increasing adaptability of Service Oriented Architecture (SOAP) in cloud computing scenario has brought lot of scope to the investigation of web services attacks, however, in the current unit we will be focusing more on web application forensics rather than web services. However, we will have little discussion on web services forensics as well here.

15.2.1 Web services forensics

The term "Web services" describes a standardized way of integrating Web-based applications using the XML, SOAP, WSDL and UDDI open standards over an Internet protocol backbone. XML is used to tag the data, SOAP is used to transfer the data, WSDL is used for describing the services available and UDDI lists what services are available.

A Web service is a method of communication between two electronic devices over a network. It is a software function provided at a network address over the Web with the service always on as in the concept of utility computing. Two conceptual elements underlie current web services:

A. Use of XML (eXtensibleMarkup Language), SOAP (Simple Object Access Protocol), and WSDL (Web Service Definition Language) as basic building material;

B. Complex applications built upon long-running transactions that are composed of other web services.

XML format underlies the entire web service architecture and its artefacts. All schemas, definition files, messages transmitted are formed by the means of XML. WSDL, a XML based definition file, defines the interface of a web service in order for the service to be invoked by other services in accordance with the specifications of internal executions. SOAP, a XML based protocol, defines the metadata of the messages to be exchanged between services. WSDL documents define operations; and they are the only mechanisms in order for web services to communicate with each other. Web services use SOAP messages by exchanging them as incoming and outgoing messages through the operations.

There are many attacks on web services, such as WSDL/UDDI scanning, parameter tampering, replays, XML rewriting, man-in-the-middle, eavesdropping, routing detours etc.

As in a document by NIST [csrc.nist.gov/publications/nistir/.../nistir-7559_forensics-web-services.pdf] we need to provide two features into web services forensics:

a) Pairwise evidence generation
b) Comprehensive evidence generation

Pair-wise evidence generation: Collect transactional evidence that occur between pairs of services at service invocation times.

Comprehensive evidence generation: On demand, compose pairs of transactional evidence collected at service invocation times and reveal global views of complex transactional scenarios that occurred during specified periods, and provide them for forensic examiners.

15.2.2 Web Application Forensics

The Major tasks an investigator needs to do while performing web application forensics are:

a) Preliminary analysis: where, we need to focus on evidence collection and protection which are in form of logs. Apart from this the investigator needs to build in confidence by using robust supporting forensic tools. Above all it all depends upon the abilities of the investigator to procure and correlate all data for inferences and conclusion.
b) Standard methodology: methodologies that are standard are easily addressable and heard in the court of law.

15.2.3 Preliminary Analysis

15.2.3.1 Application Forensics Readiness

In this the web application should be well prepared for a forensics investigation. Major activities in it are evidence collection and evidence protection, use of supportive forensics and investigator abilities and more:

i. Evidence collection: A proper Evidence collection is to be done in order to prepare a web application for a forensics investigation. Basically, and fore mostly all the logging options of the web application are enabled so as to collect maximum digital evidences. The application logs have to be set according to the case requirement and not be left in default mode which are very basic and might not log important aspects.

ii. Evidence protection: Log files are the main source of digital evidence, hence, proper mechanisms must be incorporated in order to protect these logs files and ensure that these are digitally procured and signed to be presented in court as evidence. This will certainly guarantee the accuracy of the digital evidences provided. Log files can be protected using actions like setting permissions of log files, ensuring out of reach of these log files from the hackers and following checksums to ensure integrity.

iii. Supportive forensics: Mere collection of logs will not help, we must see that these logs are supported by forensics tools evidence gathering. That is, forensic tools can help gather that information which might not be recorded in any application logs. Network or an operating system forensics tools or a third-party extra logging facilities can be utilized to achieve this.

iv. Forensics investigator abilities: The forensics investigator must have a sound knowledge and understanding of web application and its architecture etc. better understanding of security aspects and issues pertaining these applications will be required to have a forethought approach in cracking the case.

15.2.3.2 Methodology

Certain prescribed standard methodologies do exist in investigation of web application and these needs to be followed. The cruxes of these standard methodologies are:

a) Protect the web application (could be several servers) during the forensic examination so that their logs etc. can’t be modified.

b) Extract all evidence files needed for the forensics investigation:
   - Web servers and application servers logs.
   - Server side scripts which are used by the web application.
   - Web servers and application servers configuration files.
   - All third party software log files.
   - Operating system log files.

c) After collecting the files we need to perform aanalysis of those files to determine the sequence of events and the aspects where security was compromised. One way of carrying out analysis is to divide the log files according to user sessions,by doing this we will be able to remove distortions and can confine to the culprit’s sessions. Fingerprints of a web application security attack needs to be explored. The following are examples of fingerprints and patterns left by web application hacking attempts:
   a. Unusual entries in the Logs (GET requests to ASP pages which normally receive POST requests).
   b. Script abuse (CMD.exe, Root.exe, Upload. ASP).
   c. Excessive attempts from the same IP address.
   d. Unusually long processing times (SQL Injection attempt)

e. Files created or modified around the time of the suspected attack. etc.

d) Prepare a report based on the data extracted from the web application logs and other aspects.

15.2.4 Website traffic analysis

Website traffic analysis is produced by grouping and aggregating various data items captured by the web server in the form of log files while the website visitor is browsing the website. Some of the most commonly used website traffic analysis terms are listed below:

URL - A Uniform Resource Locator (URL) uniquely identifies the resource requested by the user's browser.

Hit - Each HTTP request submitted by the browser is counted as one hit. Note that HTTP requests may be submitted for non-existent content, in which case they still will be counted. For example, if one of the five image files referred by the example page mentioned above is missing, the web server will still count six HTTP requests, but in this case, five will be marked as successful (one HTML file and four images) and one as a failed request (the missing image)

Page - A page is a successful HTTP request for a resource that constitutes primary website's content. Pages are usually identified by a file extension (e.g. .html, .php, .asp, etc.) or by a missing extension, in which case the subject of the HTTP request is considered a directory and the default page for this directory is served.

File - Each successful HTTP request is counted as a file.

Visitor - A visitor is the actual person browsing the website. A typical website serves content to anonymous visitors and cannot associate visitors with the actual person browsing the website. Visitor identification may be based on their IP address or an HTTP cookie. The former approach is simple to implement, but results in all visitors browsing the same website from behind a firewall counted as a single visitor. The latter approach requires special configuration of the web server (i.e. to log HTTP cookies) and is more expensive to implement. Note that neither of the approaches identifies the actual person browsing the website and neither provides 100% accuracy in determining that the same visitor has visited the website again.

Visit - A visit is a series of HTTP requests submitted by a visitor with the maximum time between requests not exceeding a certain amount configured by the webmaster, which is typically set at 30 minutes. For example, if a visitor requested page A, then in 10 minutes page B and then in 40 minutes page C, then this visitor has generated two visits, one when pages A and B were requested and another when the page C was requested.

Host - In general, a host is the visitor's machine running the browser. Hosts are often identified by IP addresses or domain names. Those web traffic analysis tools that use

IP addresses to identify visitors use the words hosts, domain names and IP addresses interchangeably.

User Agent - User agent is a synonym for a web browser.

15.3 WEB APPLICATION FORENSICS TOOLS

As discussed in the previous section, it is very important to have supportive tools of forensics in order to have better grasp over forensics of web applications. Tools that are useful for web application forensics are Microsoft LogParser, EventLogAnalyzer, Http-analyze, Pyflag, Analog, Open Web Analytics, Mywebalizer, CORE Wisdom, Logjam, Sawmill, and Lire

15.3.1 Logparser

logparser is a flexible command line utility that was initially written by Gabriele Giuseppini, a Microsoft employee, to automate tests for IIS logging. It was intended for use with the Windows operating system, and was included with the IIS 6.0 Resource Kit Tools. The default behaviour of logparser works like a "data processing pipeline", by taking an SQL expression on the command line, and outputting the lines containing matches for the SQL expression.

Microsoft describes Logparser as a powerful, versatile tool that provides universal query access to text-based data such as log files, XML files and CSV files, as well as key data sources on the Windows operating system such as the Event Log, the Registry, the file system, and Active Directory. The results of the input query can be custom-formatted in text-based output, or they can be persisted to more specialty targets like SQL, SYSLOG, or a chart. Logparser has been also discussed in previous chapters.

15.3.2 EventLog Analyzer

Event log analysis is used for pattern matching, filtering of event occurrences, and aggregation of event occurrences into composite event occurrences. Commonly, dynamic programming strategies from algorithms are employed to save results of previous analyses for future use, since, for example, the same pattern may be match with the same event occurrences in several consecutive analysis processing. EventLog Analyzer provides the most cost-effective Security Information and Event Management (SIEM)software on the market. Using this Log Analyzer software, organizations can automate the entire process of managing terabytes of machine generated logs by collecting, analyzing, correlating, searching, reporting, andarchiving from one central location. This event log analyzer software helps to monitor file integrity, conduct log forensics analysis, monitor privileged users and comply to different compliance regulatory bodies by intelligently analyzing your logs and instantly generating a variety of reports like user activity reports, historical trend reports, and more.

15.3.3 Web log Analyzer

Web log analysis software (also called a web log analyzer) is a kind of web analytics software that passes a server log file from a web server, and based on the values contained in the log file, derives indicators about when, how, and by whom a web server is visited. Usually, reports are generated from the log files immediately, but the log files can alternatively be passed for a database and reports generated on demand.

15.3.4 Open Web Analytics[8]

Open Web Analytics (OWA) is open source web analytics software created by Peter Adams. OWA is written in PHP and uses a MySQL database, which makes it compatible for running with an AMP solution stack on various web servers. OWA is comparable to Google Analytics, though OWA is server software anyone can install and run on their own host, while Google Analytics is asoftware service offered by Google. OWA supports tracking with WordPress and MediaWiki, two popular web site frameworks. This application helps you keep track of and observe the influx of views on your website. The program also tracks your competitors and their company's growth compared to yours.

15.3.5 Webalizer

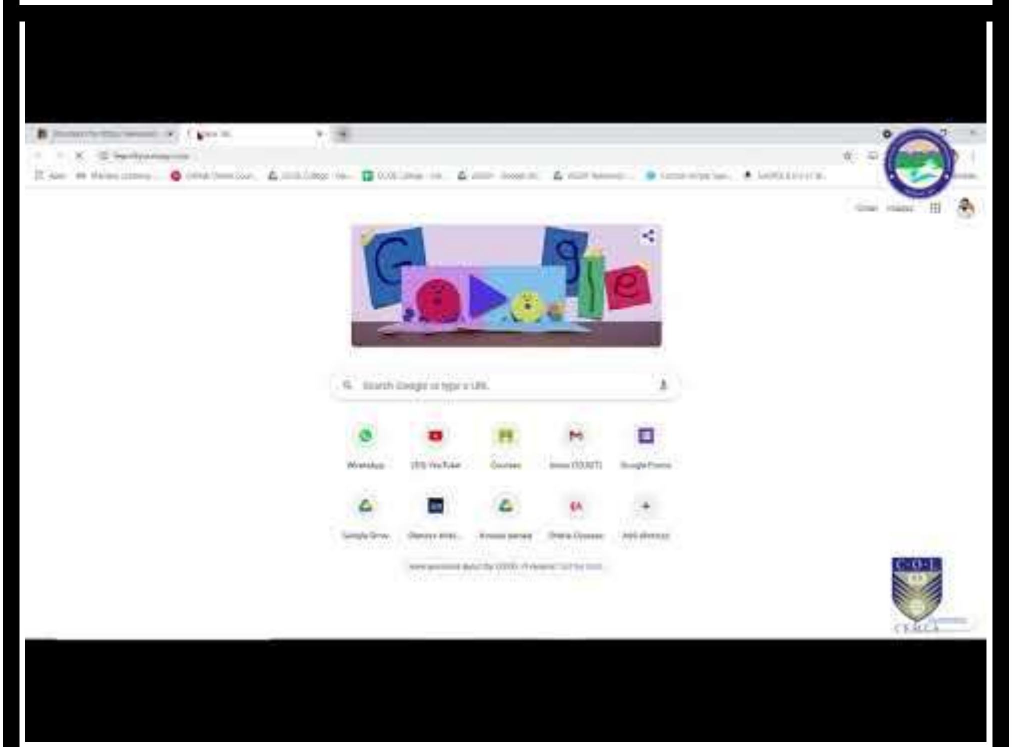
The **Webalizer** is a GPL application that generates web pages of analysis, from access and usage logs, i.e. it is web log analysis software. It is one of the most commonly used web server administration tools. It was initiated by Bradford L. Barrett in 1997. Statistics commonly reported by Webalizer include hits, visits, referrers, the visitors' countries, and the amount of data downloaded. These statistics can be viewed graphically and presented by different time frames, such as by day, hour, or month.

15.4 SUMMARY

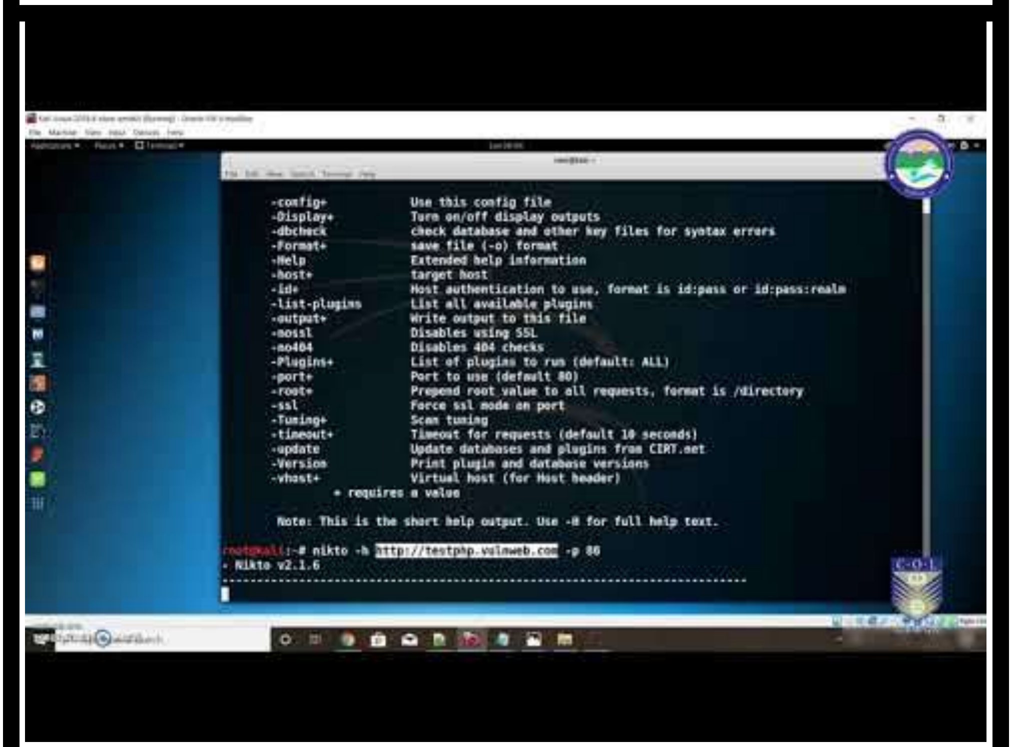
1. There can be very critical information managed through web applications nowadays which needs to handle this information with utmost care and security.
2. Cyber-attacks have become increasingly sophisticated and dangerous. Cyber-attacks have evolved into Cyber Warfare and cyber terrorism.
3. Various forms of cyber-attacks are, Spoofing, Repudiation, Privacy attacks, Denial of Service, Privilege escalation, SQL injection attacks.
4. The aspects of investigation into web attacks can be viewed in two areas; web application forensics and web services forensics.
5. Web services forensics involves pair-wise and comprehensive evidence generation. Web application forensics involves preliminary analysis and standard methodologies.
6. Tools that are useful for web application forensics are Microsoft LogParser, EventLogAnalyzer, Http-analyze, Pyflag, Analog, Open Web Analytics etc.

15.5 PRACTICAL: LET’S DO IT

VIDEO LECTURE



VIDEO LECTURE



```
    -config+            Use this config file
    -Display+           Turn on/off display outputs
    -dbcheck            check database and other key files for syntax errors
    -Format+            save file (-o) format
    -Help               Extended help information
    -host+              target host
    -id+                Host authentication to use, format is id:pass or id:pass:realm
    -list-plugins       List all available plugins
    -output+            Write output to this file
    -nossl              Disables using SSL
    -no404              Disables 404 checks
    -Plugins+           List of plugins to run (default: ALL)
    -port+              Port to use (default 80)
    -root+              Prepend root value to all requests, format is /directory
    -ssl                Force ssl mode on port
    -Tuning+            Scan tuning
    -timeout+           Timeout for requests (default 10 seconds)
    -update             Update databases and plugins from CIRT.net
    -Version            Print plugin and database versions
    -vhost+             Virtual host (for Host header)
             + requires a value

    Note: This is the short help output. Use -H for full help text.

root@kali:~# nikto -h http://testphp.vulnweb.com -p 80
- Nikto v2.1.6
---------------------------------------------------------------------------
```

VIDEO LECTURE

VIDEO LECTURE

15.6 CHECK YOUR PROGRESS

1. Fill in the blanks.

   a) _______________ is where a lower privilege user or application accesses functions or content reserved for higher privilege users or applications.
   b) Application Forensics Readiness involves _________, ___________, _____________ and ___________.

2. State True or False

   a) Open Web Analytics is written in PHP and uses a MySQL database.
   b) A distributed denial-of-service (DDoS) attack occurs when multiple systems flood the bandwidth or resources of a targeted system, usually one or more web servers

15.7 ANSWERS TO CHECK YOUR PROGRESS

1. Fill in the blanks.

   a)
   b) Vertical privilege escalation.
   c) evidence collection, evidence protection, use of supportive forensics and investigator abilities.

2. State True or False

   a) True.
   b) True.

15.8 FURTHER READINGS

1. Computer Forensics: Investigating Network Intrusions and Cyber Crime, EC-Council, Cengage learning 2010.
2. KeyunRuan, Cybercrime and Cloud Forensics: Applications for Investigation Processes, IGI Global, 2013.
3. Gutiérrez, Carlos A., Web Services Security Development and Architecture: Theoretical and Practical issues, IGI Global, 2010.
4. Linda Volonino, Reynaldo Anzaldua; Computer Forensics For Dummies, Wiley Publishing, Inc.
5. Amor Lazzez, ThabetSlimani Forensics Investigation of Web Application Security Attacks I. J. Computer Network and Information Security, 2015, 3, 10-17

15.9 MODEL QUESTIONS

1. Describe the major types of web attacks in brief.
2. What do you mean by Application Forensics Readiness?
3. Describe any 3-web application forensic tools.

References, Article Source & Contributors

[1] Cyber-attack - Wikipedia, the free encyclopedia, https://en.wikipedia.org/wiki/Cyber-attack

[2] Denial-of-service attack - Wikipedia, the free encyclopedia, https://en.wikipedia.org/wiki/Denial-of-service_attack

[3] Email spoofing - Wikipedia, the free encyclopedia, https://en.wikipedia.org/wiki/Email_spoofing

[4] Event monitoring - Wikipedia, the free encyclopedia, https://en.wikipedia.org/wiki/Event_monitoring

[5] Forensics Web Services - NIST Computer Security, csrc.nist.gov/publications/nistir/.../nistir-7559_forensics-web-services.pdf

[6] Open Web Analytics - Wikipedia, the free encyclopedia, https://en.wikipedia.org/wiki/Open_Web_Analytics

[7] Privacy issues of social networking sites - Wikipedia, https://en.wikipedia.org/wiki/Privacy_issues_of_social_networking_sites

[8] Privilege escalation - Wikipedia, the free encyclopedia, https://en.wikipedia.org/wiki/Privilege_escalation

[9] SQL injection - Wikipedia, the free encyclopedia, https://en.wikipedia.org/wiki/SQL_injection

[10] Types of Attacks, “Ethical hacking Tips”, Go4Expert, www.go4expert.com

[11] Web log analysis software - Wikipedia, the free encyclopedia, https://en.wikipedia.org/wiki/Web_log_analysis_software

[12] Web service - Wikipedia, the free encyclopedia, https://en.wikipedia.org/wiki/Web_service

[13] Webalizer - Wikipedia, the free encyclopedia, https://en.wikipedia.org/wiki/Webalizer

EXPERT PANEL

Dr. Jeetendra Pande, Associate Professor- Computer Science, School of Computer Science & IT, Uttarakhand Open University, Haldwani

Dr. Ajay Prasad, Sr. Associate Professor, University of Petroleum and Energy Studies, Dehradun

Dr. Akashdeep Bharadwaj, Professor, University of Petroleum and Energy Studies, Dehradun

Mr. Sridhar Chandramohan Iyer, Assistant Professor- Universal College of Engineering, Kaman, Vasai, University of Mumbai

Mr. Rishikesh Ojha, Digital Forensics and eDiscovery Expert

Ms. Priyanka Tewari, IT Consultant

Mr. Ketan Joglekar, Assistant Professor, GJ College, Maharastra

Dr. Ashutosh Kumar Bhatt, Associate Professor, Uttarakhand Open University, Haldwani

Dr. Sangram Panigrahi, Assistant Professor, Siksha 'O' Anusandhan, Bhubaneswar

UTTARAKHAND OPEN UNIVERSITY

This MOOC has been prepared with the support of

C·O·L

CEMCA