Forensic Readiness

Be ready for the incident

Forensic Readiness

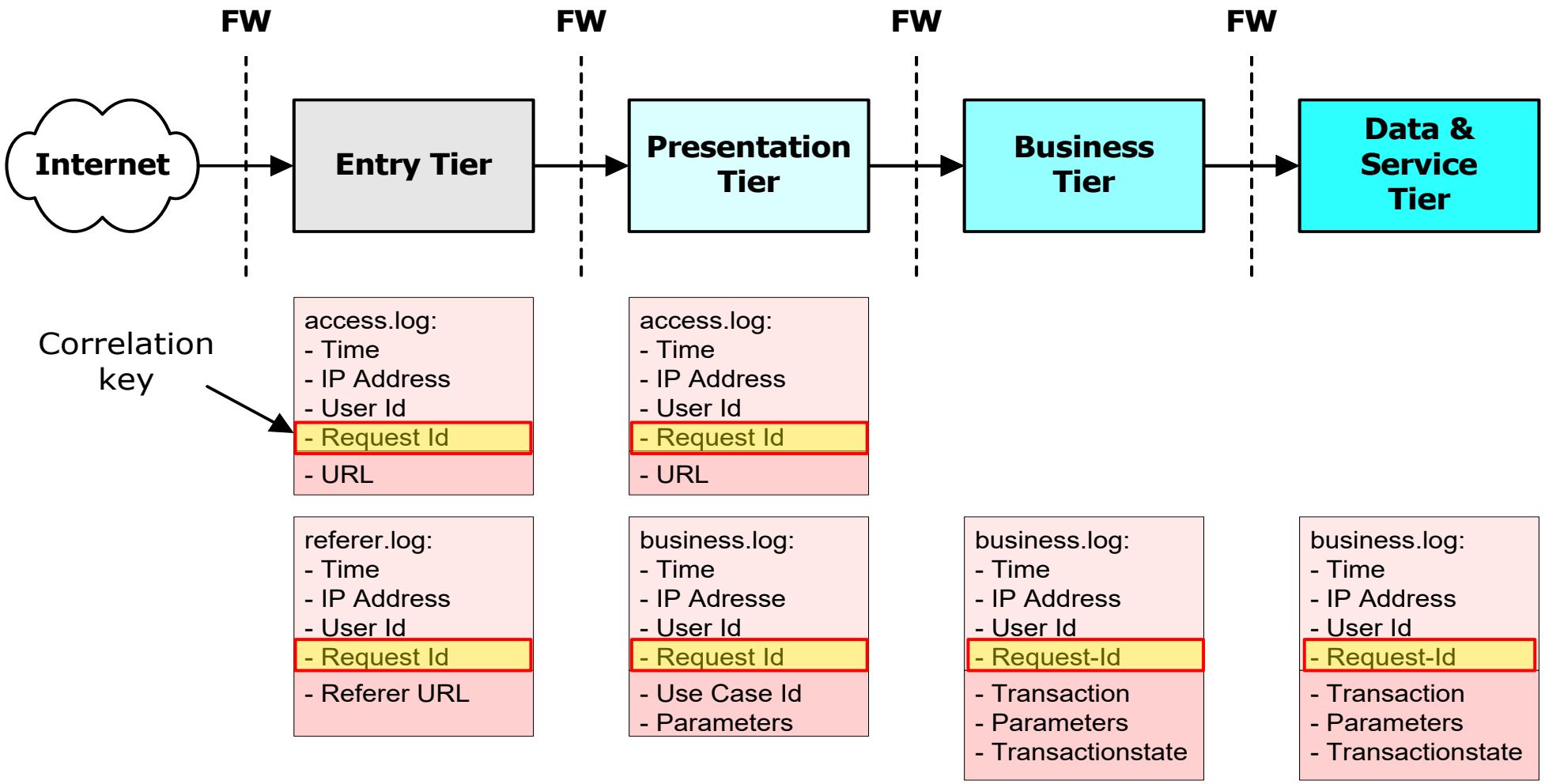
According to Forensics Readiness Guidelines (NICS, 2011)

Forensic Readiness is *having an appropriate level of capability in order to be able to preserve, collect, protect and analyze digital evidence so that this evidence can be used effectively: in any legal matters; in security investigations; in disciplinary proceeding; in an employment tribunal; or in a court of law*.

In other words: “Make sure you can correlate events from different it systems in a cascaded multi-tier and micro-service architecture”.

Web Forensic Readiness

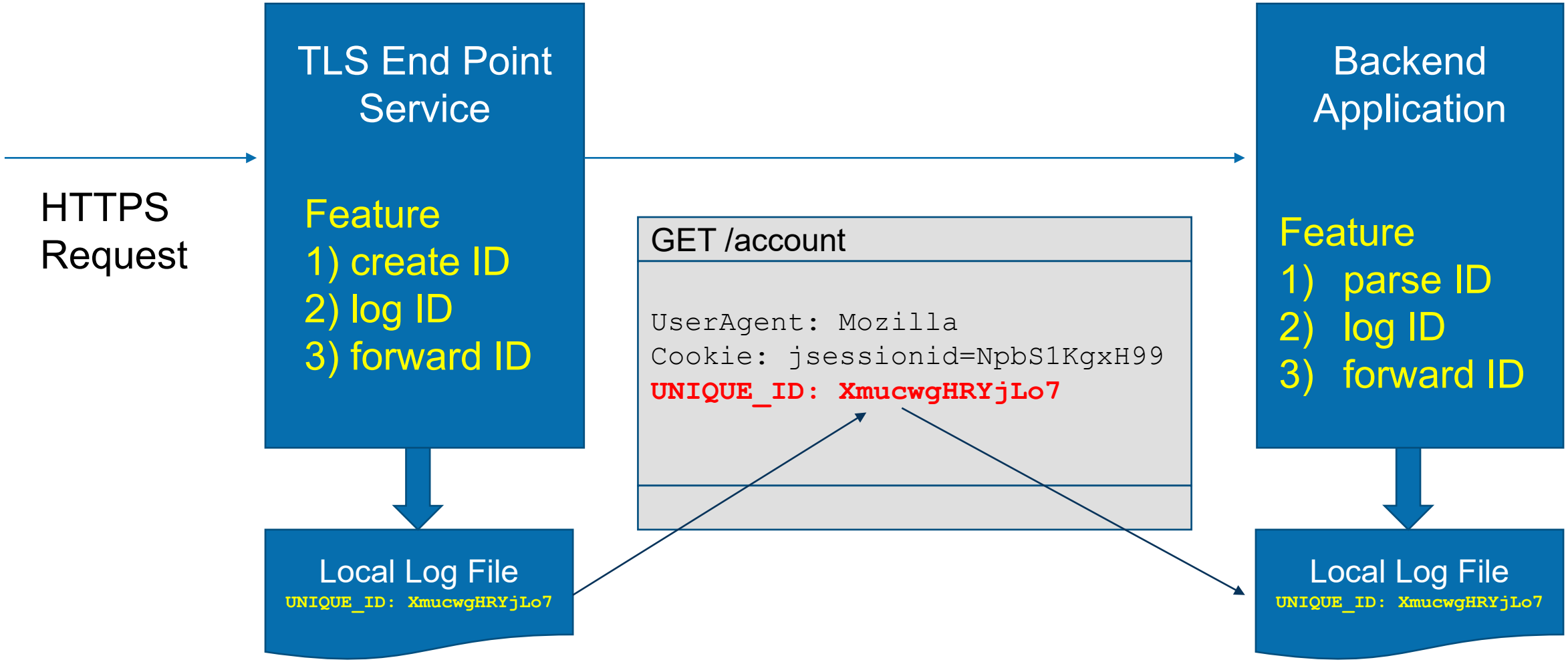
Forensic Readiness with UniqueID == RequestID



Add Unique-ID to Requests

mod_headers

Adding UNIQUE-ID into Request Header to Backend Application



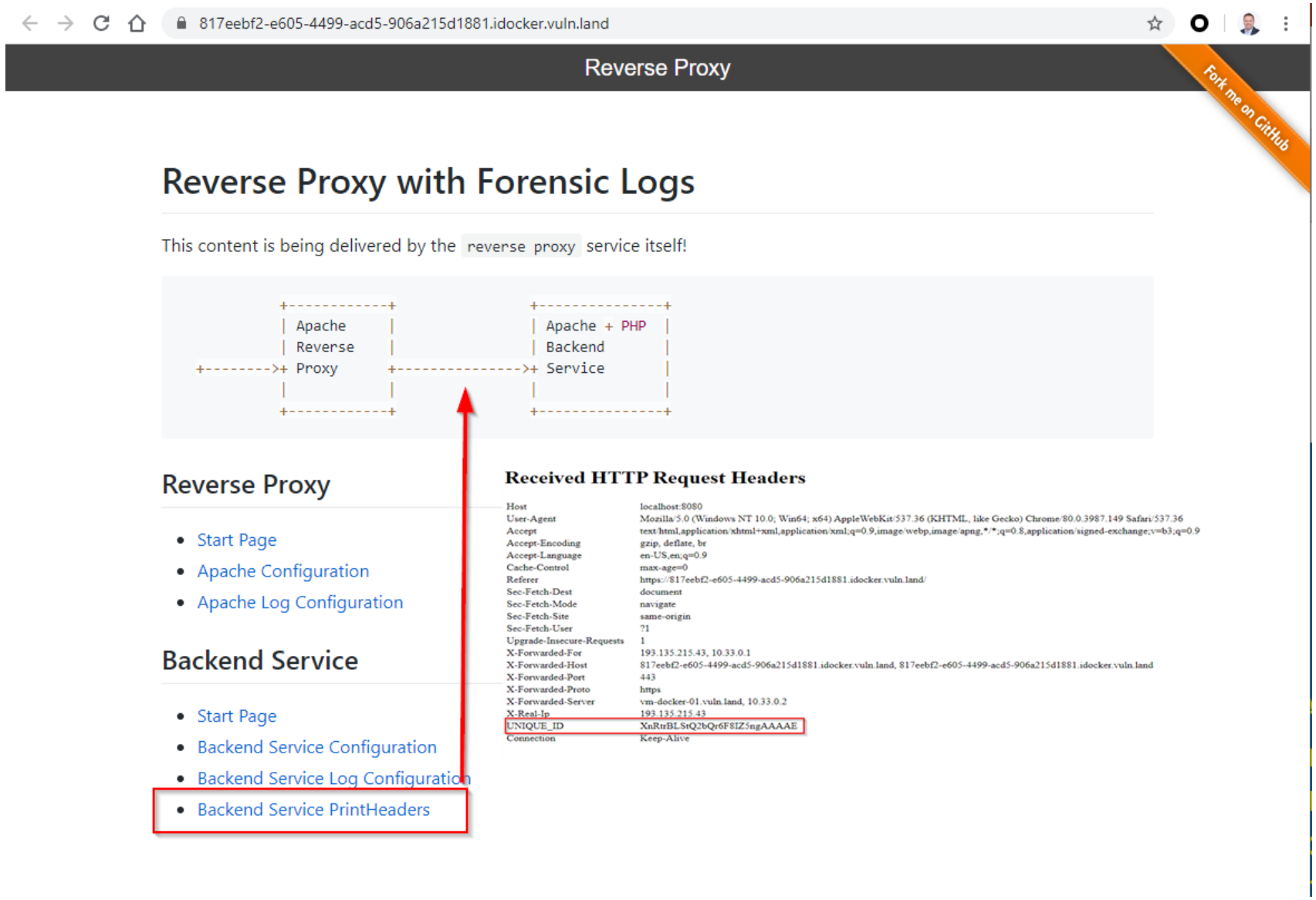
Adding Headers to Request (mod_headers)

RequestHeader append UNIQUE_ID '%{UNIQUE_ID}e'

Format	Description
%%	The percent sign
%t	The time the request was received in Universal Coordinated Time since the epoch (Jan. 1, 1970) measured in microseconds. The value is preceded by t=.
%D	The time from when the request was received to the time the headers are sent on the wire. This is a measure of the duration of the request. The value is preceded by D=. The value is measured in microseconds.
%l	The current load averages of the actual server itself. It is designed to expose the values obtained by getloadavg() and this represents the current load average, the 5 minute average, and the 15 minute average. The value is preceded by l= with each average separated by /. Available in 2.4.4 and later.
%i	The current idle percentage of httpd (0 to 100) based on available processes and threads. The value is preceded by i=. Available in 2.4.4 and later.
%b	The current busy percentage of httpd (0 to 100) based on available processes and threads. The value is preceded by b=. Available in 2.4.4 and later.
%{VARNAME}e	The contents of the environment variable VARNAME.
%{VARNAME}s	The contents of the SSL environment variable VARNAME, if mod_ssl is enabled.

https://httpd.apache.org/docs/current/mod/mod_headers.html

Screenshot from the Hacking-Lab Exercise



Clicking on «Backend Service PrintHeaders» on can see the Request between the Reverse Proxy and the Backend Service

This is very usefull to see, if the Reverse Proxy is really sending the UNIQUE-ID

Screenshot from the Hacking-Lab Exercise

8116bd66-b236-4bf7-9fc9-29686c6bdfd7.idocker.vuln.land/backend/printheader.php

Received HTTP Request Headers

Host	localhost:8080
User-Agent	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/80.0.3987.132 Safari/537.36
Accept	text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
Accept-Encoding	gzip, deflate, br
Accept-Language	en-US,en;q=0.9
Referer	https://8116bd66-b236-4bf7-9fc9-29686c6bdfd7.idocker.vuln.land/
Sec-Fetch-Dest	document
Sec-Fetch-Mode	navigate
Sec-Fetch-Site	same-origin
Sec-Fetch-User	?1
Upgrade-Insecure-Requests	1
X-Forwarded-For	62.2.85.146, 10.197.0.1
X-Forwarded-Host	8116bd66-b236-4bf7-9fc9-29686c6bdfd7.idocker.vuln.land, 8116bd66-b236-4bf7-9fc9-29686c6bdfd7.idocker.vuln.land
X-Forwarded-Port	443
X-Forwarded-Proto	https
X-Forwarded-Server	vm-docker-01.vuln.land, 10.197.0.4
X-Real-Ip	62.2.85.146
UNIQUE_ID	XmucwgHRYjLo7yj4klPB5gAAAAI
Connection	Keep-Alive

Unique-ID

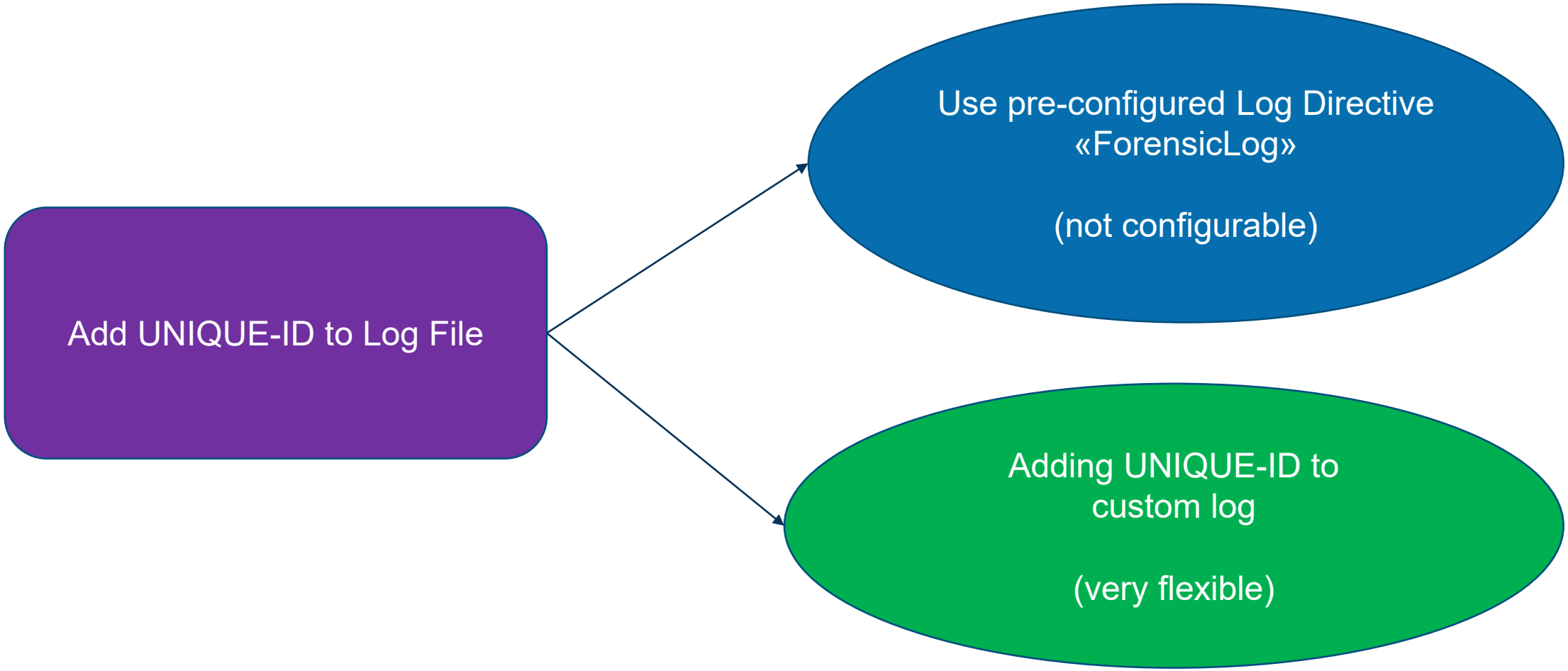
Navigation

Go back to backend service or to reverse proxy

Add Unique-ID to Logfile

Apache Web Server

Apache Web Server: two Types of Log Configurations



ForensicLog with Unique-ID for Apache Web Servers

httpd.conf

```
LoadModule log_forensic_module modules/mod_log_forensic.so
LoadModule unique_id_module modules/mod_unique_id.so
LogFormat "%{UNIQUE_ID}e %h %l %u %t \"%r\" %>s %b" forensic
CustomLog logs/forensic.log forensic
ForensicLog logs/forensiclog.log
```

mod_log_forensic

mod_unique_id

(log format is configurable)
mod_unique_id

mod_log_forensic
(fixed format)

Static «ForensicLog»

Apache Web Server

ForensicLog with «mod_log_forensic»

mod_log_forensic
(fixed format)

mod_log_forensic - Apache HTTP

httpd.apache.org/docs/2.4/mod/mod_log_forensic.html

APACHE HTTP SERVER PROJECT

Modules | Di

Apache HTTP Server Version 2.4

Apache > HTTP Server > Documentation > Version 2.4 > Modules

Apache Module mod_log_forensic

Available Languages: en | fr | ja | tr

Description:	Forensic Logging of the requests made to the server
Status:	Extension
Module Identifier:	log_forensic_module
Source File:	mod_log_forensic.c
Compatibility:	mod_unique_id is no longer required since version 2.1

Configuration Directive:

ForensicLog logs/forensiclog.log

Summary

This module provides for forensic logging of client requests. Logging is done before and after processing a request, so the forensic log contains two log lines for each request. The forensic logger is very strict, which means:

- The format is fixed. You cannot modify the logging format at runtime.
- If it cannot write its data, the child process exits immediately and may dump core (depending on your CoreDumpDirectory configuration).

The check_forensic script, which can be found in the distribution's support directory, may be helpful in evaluating the forensic log output.

ForensicLog with «mod_log_forensic»

mod_log_forensic
(fixed format)

ForensicLog logs/forensiclog.log

root@sully:/opt/applic/httpd/logs

```
+XmhUs38AAAEAAGu6wkgAAABB|POST /cron/vmcontrol.html?job=updateList HTTP/1.1|Accept-Encoding:identity|Content-Length:452|Host:www.hacking-lab.com|
Content-Type:application/x-www-form-urlencoded|Connection:close|User-Agent:Python-urllib/2.7
-XmhUs38AAAEAAGu6wkgAAABB
+XmhU7H8AAAEAAGuTl9kAAAAE|POST /cron/vmcontrol.html?job=updateList HTTP/1.1|Accept-Encoding:identity|Content-Length:1899|Host:www.hacking-lab.com
|Content-Type:application/x-www-form-urlencoded|Connection:close|User-Agent:Python-urllib/2.7
-XmhU7H8AAAEAAGuTl9kAAAAE
+XmhVBX8AAAEAAGu6wkkAAABC|GET /user/profile/dotton/?__cookie_try=1 HTTP/1.1|Host:www.hacking-lab.com|AMP-Cache-Transform:google;v="1..3"|Connecti
on:keep-alive|Accept:text/html,application/xhtml+xml,application/signed-exchange;v=b3,application/xml;q=0.9,*/*;q=0.8|From:googlebot(at)googlebot
.com|User-Agent:Mozilla/5.0 (Linux; Android 6.0.1; Nexus 5X Build/MMB29P) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/41.0.2272.96 Mobile Safar
i/537.36 (compatible; Googlebot/2.1; +http%3a//www.google.com/bot.html)|Accept-Encoding:gzip,deflate,br
-XmhVBX8AAAEAAGu6wkkAAABC
+XmhVBn8AAAEAAGu6wkoAAABC|GET /user/profile/dotton/?__cookie_try=1 HTTP/1.1|Host:www.hacking-lab.com|AMP-Cache-Transform:google;v="1..3"|Cookie:H
LSSL=l2XaLrcY2hRufAHX0B5AzcRQSBASJrAy|Connection:keep-alive|Accept:text/html,application/xhtml+xml,application/signed-exchange;v=b3,application/x
ml;q=0.9,*/*;q=0.8|From:googlebot(at)googlebot.com|User-Agent:Mozilla/5.0 (Linux; Android 6.0.1; Nexus 5X Build/MMB29P) AppleWebKit/537.36 (KHTML
, like Gecko) Chrome/41.0.2272.96 Mobile Safari/537.36 (compatible; Googlebot/2.1; +http%3a//www.google.com/bot.html)|Accept-Encoding:gzip,deflat
e,br
-XmhVBn8AAAEAAGu6wkoAAABC
+XmhVDH8AAAEAAGuTl9oAAAAF|GET /robots.txt HTTP/1.1|Host:www.hacking-lab.com|Connection:keep-alive|User-Agent:Mozilla/5.0 (Windows NT 10.0; Win64;
 x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/80.0.3987.132 Safari/537.36|DNT:1|Sec-Fetch-Dest:empty|Accept:*/*|Sec-Fetch-Site:none|Sec-Fet
ch-Mode:no-cors|Accept-Encoding:gzip, deflate, br|Accept-Language:zh-CN,zh;q=0.9,en-GB;q=0.8,en;q=0.7
-XmhVDH8AAAEAAGuTl9oAAAAF
+XmhVDH8AAAEAAGu6wksAAABD|GET / HTTP/1.1|Host:www.hacking-lab.com|Connection:keep-alive|DNT:1|Upgrade-Insecure-Requests:1|User-Agent:Mozilla/5.0
(Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/80.0.3987.132 Safari/537.36|Sec-Fetch-Dest:document|Accept:text/html,
application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9|Sec-Fetch-Site:none|Sec-Fetch-
Mode:navigate|Referer:https%3a//www.vulnhub.com/resources/|Accept-Encoding:gzip, deflate, br|Accept-Language:zh-CN,zh;q=0.9,en-GB;q=0.8,en;q=0.7
-XmhVDH8AAAEAAGu6wksAAABD
```

Custom Log «mod_log_config»

Apache Web Server

Custom Log

Custom Log Formats

The format argument to the LogFormat and CustomLog directives is a string. This string is used to log each request to the log file. It can contain literal characters copied into the log files and the C-style control characters "\n" and "\t" to represent new-lines and tabs. Literal quotes and backslashes should be escaped with backslashes.

The characteristics of the request itself are logged by placing "%" directives in the format string, which are replaced in the log file by the values as follows:

Format String	Description
%%	The percent sign.
%a	Client IP address of the request (see the mod_remoteip module).
%{c}a	Underlying peer IP address of the connection (see the mod_remoteip module).
%A	Local IP-address.
%B	Size of response in bytes, excluding HTTP headers.
%b	Size of response in bytes, excluding HTTP headers. In CLF format, *i.e.* a '–' rather than a 0 when no bytes are sent.
%{*VARNAME*}C	The contents of cookie *VARNAME* in the request sent to the server. Only version 0 cookies are fully supported.
%D	The time taken to serve the request, in microseconds.
%{*VARNAME*}e	The contents of the environment variable *VARNAME*.
%f	Filename.
%h	Remote hostname. Will log the IP address if HostnameLookups is set to Off, which is the default. If it logs the hostname for only a few hosts, you probably have access control directives mentioning them by name. See the Require host documentation.
%{c}h	Like %h, but always reports on the hostname of the underlying TCP connection and not any modifications to the remote hostname by modules like mod_remoteip.
%H	The request protocol.
%{*VARNAME*}i	The contents of *VARNAME*: header line(s) in the request sent to the server. Changes made by other modules (e.g. mod_headers) affect this. If you're interested in what the request header was prior to when most modules would have modified it, use mod_setenvif to copy the header into an internal environment variable and log that value with the %{*VARNAME*}e described above.
%k	Number of keepalive requests handled on this connection. Interesting if KeepAlive is being used, so that, for example, a '1' means the first keepalive request after the initial one, '2' the second, etc...; otherwise this is always 0 (indicating the initial request).
%l	Remote logname (from identd, if supplied). This will return a dash unless mod_ident is present and IdentityCheck is set On.
%L	The request log ID from the error log (or '-' if nothing has been logged to the error log for this request). Look for the matching error log line to see what request caused what error.
%m	The request method.
%{*VARNAME*}n	The contents of note *VARNAME* from another module.
%{*VARNAME*}o	The contents of *VARNAME*: header line(s) in the reply.
%p	The canonical port of the server serving the request.

http://httpd.apache.org/docs/current/mod/mod_log_config.html

CustomLog forensic.log

LogFormat "%{UNIQUE_ID}e %h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i\"" forensic
CustomLog logs/forensic.log forensic

root@sully:/opt/applic/httpd/logs

```
XmhUs38AAAEAAGu6wkgAAABB 212.254.246.103 - - [11/Mar/2020:04:02:11 +0100] "POST /cron/vmcontrol.html?job=updateList HTTP/1.1" 200 -
- 137.97.73.10 - - [11/Mar/2020:04:02:35 +0100] "-" 408 -
- 137.97.73.10 - - [11/Mar/2020:04:02:35 +0100] "-" 408 -
XmhU7H8AAAEAAGuTl9kAAAAE 212.254.246.102 - - [11/Mar/2020:04:03:08 +0100] "POST /cron/vmcontrol.html?job=updateList HTTP/1.1" 200 -
XmhVBX8AAAEAAGu6wkkAAABC 66.249.76.150 - - [11/Mar/2020:04:03:33 +0100] "GET /user/profile/dotton/?__cookie_try=1 HTTP/1.1" 302 247
XmhVBn8AAAEAAGu6wkoAAABC 66.249.76.150 - - [11/Mar/2020:04:03:34 +0100] "GET /user/profile/dotton/?__cookie_try=1 HTTP/1.1" 200 2421
XmhVDH8AAAEAAGuTl9oAAAAF 137.220.138.132 - - [11/Mar/2020:04:03:40 +0100] "GET /robots.txt HTTP/1.1" 302 237
XmhVDH8AAAEAAGu6wksAAABD 137.220.138.132 - - [11/Mar/2020:04:03:40 +0100] "GET / HTTP/1.1" 302 222
XmhVDH8AAAEAAGuTl9sAAAAF 137.220.138.132 - - [11/Mar/2020:04:03:40 +0100] "GET /robots.txt?__cookie_try=1 HTTP/1.1" 302 222
XmhVDH8AAAEAAGu6wkwAAABD 137.220.138.132 - - [11/Mar/2020:04:03:40 +0100] "GET /index.html HTTP/1.1" 200 9257
XmhVDX8AAAEAAGuTl9wAAAAF 137.220.138.132 - - [11/Mar/2020:04:03:41 +0100] "GET /robots.txt HTTP/1.1" 302 -
XmhVDX8AAAEAAGu6wk0AAABD 137.220.138.132 - - [11/Mar/2020:04:03:41 +0100] "GET /misc/css/hackinglab.css HTTP/1.1" 200 11343
XmhVDX8AAAEAAGu6wk4AAABD 137.220.138.132 - - [11/Mar/2020:04:03:41 +0100] "GET /export/sites/www.hacking-lab.com/robots.txt HTTP/1.1" 200 34
XmhVDX8AAAEAAGuTl90AAAAF 137.220.138.132 - - [11/Mar/2020:04:03:41 +0100] "GET /misc/js/tabs/jquery.js HTTP/1.1" 200 67887
XmhVDn8AAAEAAGu6wk8AAABD 137.220.138.132 - - [11/Mar/2020:04:03:42 +0100] "GET /misc/js/code_highlighter.js HTTP/1.1" 200 4181
XmhVDn8AAAEAAGuTl94AAAAF 137.220.138.132 - - [11/Mar/2020:04:03:42 +0100] "GET /misc/css/font-awesome/css/font-awesome.css HTTP/1.1" 200 7439
XmhVDn8AAAEAAGu6wlAAAABD 137.220.138.132 - - [11/Mar/2020:04:03:42 +0100] "GET /misc/css/animate.css HTTP/1.1" 200 4430
XmhVD38AAAEAAGu6wlEAAABD 137.220.138.132 - - [11/Mar/2020:04:03:43 +0100] "GET /misc/js/hllogin-o.js HTTP/1.1" 200 817
```

Conclusion

Forensic-Readiness with Unique-ID per Request

What is the benefit of the Unique-ID?

- Correlation of events among multi-tier and micro-service architectures
- Because a timestamp is **not sufficient**!!!!
- First server (internet facing) that should generate the unique-id and add the id to the own log files. Furthermore, the unque-id should be handed over to backend services, usually by adding a special header. The backend services should parse the unique-id and add it to the own logs. Furthermore, it should add the unique-id to any further server or instance. If this would be the case for all services – you can always find out what who, when, where something happended.
- Without the unique-id, companies are lost and must relay on timestamps