SE Labs

Post-Incident Data Log Guide

A guide to the forensic data collected during attack testing

Ref: [v.1.0]

Contents

1. Introduction

This guide is designed to help you navigate the data provided after an attack test. The data will provide enough detail to allow verification of the test results and potentially aid in discovering reasons and solutions for any failures suffered by the product(s).

SE Labs verifies test results using digital forensics, rather than relying on the alerts produced by tested products. Testers record each product's claims of detection and protection and then test these claims against data gathered during the test using a variety of tools and methods.

Whether testing endpoint security products or appliances, much data is gathered from the attacked endpoint to demonstrate whether or not the attack worked and, if so, to what extent. Logs are also collected from any non-endpoint products involved in the test, such as network appliances under test and network appliances used to monitor the test.

In addition to using this data to verify our findings, we also provide this data to enable third parties to verify results independently and potentially to improve their own products using the detailed evidence gathered. To this end we endeavour to use freely-available tools that any independent lab should be able to acquire and deploy with minimum expense.

Typically, the data gathered includes information on changes made to the target systems during and after an attack. We also record all network traffic entering and leaving the target, as well as real-time events from the target including thread and process-related activities.

Full network traffic dumps are recorded for troubleshooting purposes and are not publicly available, although parts may be disclosed when investigating some technical issues with products.

SE Labs specialises in testing using exploit-based threats, which are collected or generated; used; and stored in a format that allows them to be replicated in other lab environments.

2. Client-side logs

The following data is collected from endpoint target systems:

2.1. System changes (file system; Windows Registry; Windows Services)

- Files: added, removed, modified
- Registry keys/values added, removed, modified
- Detailed execution metadata from Windows Prefetch files
- Hidden processes (using 'rootkit' techniques)

2.2. System activity

- Complete Windows event logging

2.3. Network activity

- Complete packet capture (on the client)

2.4. Endpoint production product logs

- Logs generated by any available endpoint protection products running at the time of the test. Optionally, extra debugging information may be collected.

2.5. Logging tools

These logs are generated using tools including:

2.5.1. Regshot (http://sourceforge.net/projects/regshot/) – file system and Registry changes
2.5.2. Bespoke scripts – Windows Services changes
2.5.3. WinPrefetchView (http://www.nirsoft.net/utils/win_prefetch_view.html) – Windows Prefetch file information
2.5.4. MoonSols DumpIt (http://www.moonsols.com/2011/07/18/moonsols-dumpit-goes-mainstream/) or similar in combination with The Volatility Framework (https://code.google.com/archive/p/volatility/) – detect hidden processes
2.5.5. Process Monitor (https://technet.microsoft.com/en-us/sysinternals/processmonitor.aspx) – Windows event logging
2.5.6. Wireshark (https://www.wireshark.org/) – client-side network packet capture

3. Server-side logs

The following data is collected from tested network appliances and other technologies involved directly in the test:

3.1. Appliance detection logs

These logs are generated directly by the appliance(s) under test and other network systems deployed to monitor network traffic.

3.2. Appliance configuration/diagnostic files

Data demonstrating the operating system version, license(s) status and configuration settings.

4. Attack replays

Web (HTTP/S) sessions for attacks and, in some cases, non-malicious web sessions are stored in a replayable format. When used to replay attacks this allows third-party labs to replicate the same attack as that used in the test, including the same exploit and payload stages. When replaying targeted attacks, the system reduces the effort needed by the lab to set up the attack environment and ensures that the attack runs in exactly the same way as it did during the test.

Replayable web sessions are created and replayed using Fiddler2 (https://www.telerik.com/download/fiddler/fiddler2).

5. Miscellaneous

During a test various interesting files, such as payloads and other generated or downloaded data, may appear on the network and/or endpoints. At our discretion we may collect and provide copies of this data separately for convenience, although it is usually possible to extract such data from the packet captures, memory dumps and Windows event logging files.

6. Understanding client-side logs

Logs taken directly from the target clients are provided in sets, with a single archive file containing a full set of logs from one target client and for one specific incident.

For example, in a test of endpoint products that comprises 100 targeted attacks there should be 100 archives for each product named according to the following convention: ***nnn_pid.zip*** (e.g. 056_EXA.zip).

In this example we have a bundle of logs and other data that was produced on the target client system running the fictional Example Protection product when attacked by the 56th threat.

The typical contents of such an archive will look like the following*:

Description	**Filename (approx.)**	**Notes**
File system and Registry changes	056_EXA_clean.hivu	.hivu files are database binary files, while .txt files are exported reports containing human-readable entries.
	056_EXA_exposed.hivu	
	056_EXA_exposed.txt	
	056_EXA_scanned.hivu	
	056_EXA_scanned.txt	
Windows Services changes	all_services_clean.txt	Diff. these files to see the changes.
	all_services_exposed.txt	
Windows logs	Event Viewer	
Endpoint product's debugging logs (optional)	ABC123.X.log	
	ABC123.Y.log	
	ABC123.Z.log	
Endpoint products default logs	EXA_a.log	
	EXA_b.log	
Packet capture from client	net.cap	
Windows Prefetch metadata	Prefetch_details.txt	
Windows event logging	ProcessMonitor.PML	
Dropped files (optional)	File.jpg.exe.zip	Stored in password-protected archives, password = infected
	Tmp.js.zip	

* If 'rootkit' techniques are found to be involved in the attack then a full system memory dump will also be included in the archive. These are very large files and are not included unless necessary.

When a network appliance test is conducted the additional log files from the appliance is included along with the above client-side logs.

7. Understanding attack replays

Web (HTTP/S) sessions for attacks and non-malicious web sessions are stored in a replayable format.

Packet capture files created on the client are useful as evidence of an event, or series of events, but they do not easily allow other labs to replicate those events. Web replay files, on the other hand, allow third parties to re-test to verify results and potentially improve products that were initially unable to detect or repel the attack.

Each attack requires one replay file so in a test that includes 100 attacks there should be 100 replay files named according to the following convention: ***{path}/nnn/url.zip*** (e.g. /tmp/056/bad-site.com.zip).

In this example we have the 56th replayable attack of the test. This was found to exist on the internet, at the time of testing, at the (fictional) URL www.bad-site.com.

Load the archive file into Fiddler2 to view the internal session data formatted in a clear manner and ready to be used in a replayed test. Use Fiddler2 as a proxy internet connection for the target system.

The content of the archive includes a directory called 'raw' and a file called _index.htm. The _index.htm file contains a list of URLs involved in the entire session and a list of files that comprise the client's requests and the servers' responses. These files are stored in the 'raw' directory.

The contents of the _index.htm file will look similar to the text below:

			#	**Result**	**Protocol**	**Host**	
C	S	M	11	301	HTTP	bad-site.com	/
C	S	M	12	200	HTTP	www.bad-site.com	/
C	S	M	13	200	HTTP	www.bad-site.com	/wp-content/themes/g.css
C	S	M	14	200	HTTP	www.bad-site.com	/js/bad,js

The 'raw' session files will look similar to the directory listing below:

0000_c.txt
0000_m.xml
0000_s.txt
0001_c.txt
0001_m.xml
0001_s.txt

The content of these files will look similar to this:

0000_c.txt *(c=client request)*

GET http://bad-site.com/ HTTP/1.1
Accept: application/x-ms-application, image/jpeg, application/xaml+xml, image/gif, image/jpeg, application/x-ms-xbap, */*
Accept-Language: en-GB
User-Agent: Mozilla/4.0 etc…

Host: bad-site.com

0000_s.txt *(s=server response)*

HTTP/1.0 301 Moved permanently
Date: Tue, 2 Feb 2016 17:17:10 GMT
Server: Apache
X-Powered-By: PHP/5.4.19 etc…

Location: http://www.bad-site.com/