



# Analyzing Web-Servers for Malicious Content Using Monkey-Spider Honeyclient

Sept 20, 2009

Project Team: Serge Gorbunov, Sami K. Guirguis

Document Author: Serge Gorbunov

Document Editor: Sami K. Guirguis

## **Content**

Abstract .....	3
Background .....	3
Setup .....	4
Test Description .....	6
Test Results .....	6
Performance Results .....	8
Advantages vs. Disadvantages of the system .....	8
Conclusion .....	9
References .....	10

## **Abstract**

Most computer networks visit internet sites hundreds of times every day. Firewalls and antivirus systems are used to protect them from malicious attacks, yet still infecting rate are high due to a variety of reasons. Some do not update security software often enough to keep up with the new threats, or security software fails itself to identify a potential threat or a zero-day attack. In either way, if sufficient information was given to the user about the malicious web-server, the infection could have been prevented by avoiding communication with the web-server. To identify malicious web-servers a number of attempts have been made, one is by use of honeyclients. This paper includes evaluation of one such honeyclient - Monkey-Spider, which is identified as low interaction honeyclient system. In addition, it presents a few tests and improvements made to the system and discusses its advantages vs. disadvantages.

## **Background**

Honeypots – systems that passively wait to be compromised and attacked – have been around for more than a decade. One of the reasons to place a honeypot would be to study how attacks are performed, tools used, and motives behind them. However, due to the dynamic expansion of the Internet, active systems able to identify malicious web-servers are needed. Since malware on those servers is highly dynamic and kept up-to-date against the antivirus software, users often get infected even with multi-layered security level in place. In order to identify such malicious web-servers, a new system called honeyclient has been designed. Honeyclient, in contrast to honeypot, constantly scans the internet, looking for malicious pages and identifying potential threats. Similar to honeypots, honeyclients are divided into two main categories – low and high interaction. High interaction honeyclient provides a full operating system for the malware to interact with, while low interaction honeyclient does not.

## Setup

For this project, a low interaction honey client system was chosen - Monkey-Spider due to the following factors:

- Ease of setup
- It is a free honeyclient
- Applicable for distributed crawling/analysis
- Provides reasonably fast results

Monkey-Spider system consists of 3 easy steps required for webserver analysis:

- First, a user is required to generate initial seed list. For that, Monkey-Spider provides a few scripts that will help gather seeds by using yahoo/livesearch engines, blacklists, or spam emails. We extended it, by adding 2 additional scripts. One is used to generate typosquatted domain names - registered domain names which closely resemble high-traffic websites, but feature common misspellings. The second script was written to perform web search by using google's AJAX API. Both included and extended scripts are capable to generate a variety of seed lists with a high probability of finding malicious content.
- Second step consist of crawling the seeds using Heritrix webcrawler. Heritrix crawls the initial seed lists, extracts links from the headers and the content, and continues until stopped or complete.
- Final step consists of analyzing the content using ClamAV anti-virus software. After scan is complete results are entered into the database for further analysis.

Figure 1 shows the setup architecture built for this project. As a base, Windows Vista platform was used for web crawling, database and malware storage. Ubuntu 9.04 was setup as a virtual machine for malware analysis.

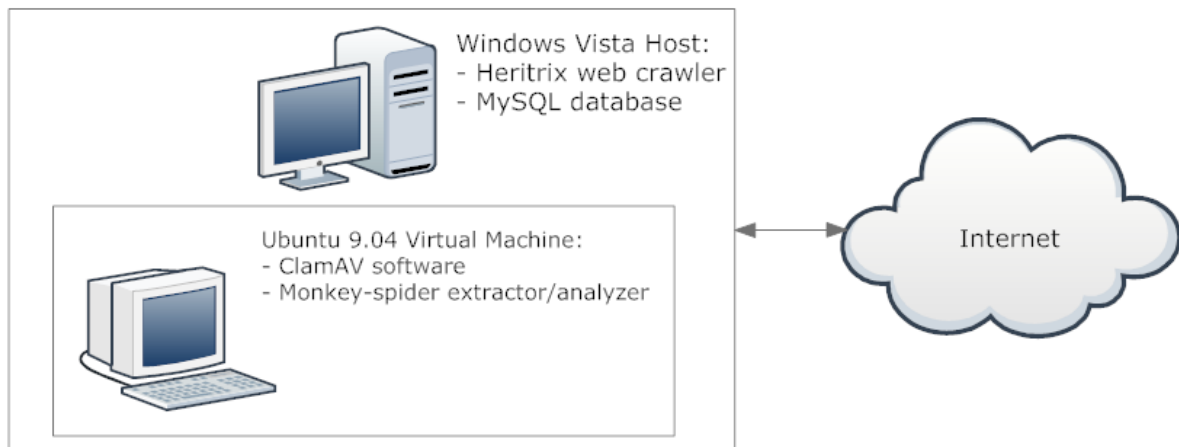


Figure 1: Setup Architecture

The workflow of the system is shown in Figure 2. Malware analyzer was contained in the Virtual Machine to prevent malware infection of the host, and control its high CPU demands using Virtual Environment Technology. We modified Monkey-Spider by extending and improving its functionality:

- Separate stand-alone scripts have been converted using Object-Oriented Design
- Improved Error handling
- Improved logging
- Changed from PostgreSQL to MySQL database
- Added functionality to log extraction and scanning times into the database
- Added scripts to analyze malware extractor/scanner statistics

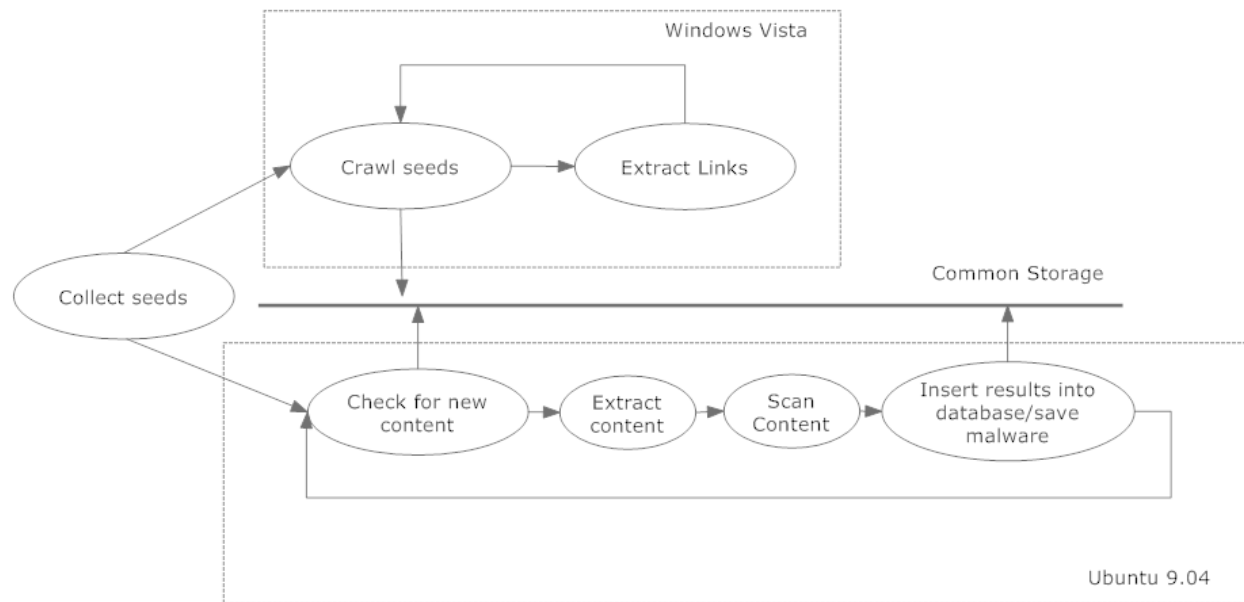


Figure 2: System Workflow

## Tests Description

For tests performed during the 3 months of this project blacklists, google search and typosquatted domain names were used mainly. Tests did not exceed 1 week in duration, due to very high internet demands. Heritrix was set to crawl “broad-but-shallow” with 100 hops level. Since Heritrix uses JAVA threads to perform crawling of each separate URI, 1000 MB was allocated to JAVA under Vista Host. Heritrix was set to use 500 threads.

## Test Results

This section presents test results performed in this project. Table 1 illustrates the top 10 infected domain names. In total, 184 unique infected domain names have been identified during the project.

Table 1: Top 10 infected domains \*

Domain Name	Number of Infections
<a href="http://www.ginedis.com">http://www.ginedis.com</a>	353
<a href="http://ginedis.com">http://ginedis.com</a>	177
<a href="http://allentownwomenscenter.com">http://allentownwomenscenter.com</a>	49
<a href="http://www.downloadteens.com">http://www.downloadteens.com</a>	47
<a href="http://baw.hotgames.cz">http://baw.hotgames.cz</a>	35
<a href="http://purplehoodie.com">http://purplehoodie.com</a>	33
<a href="http://www.minijeux.org">http://www.minijeux.org</a>	21
<a href="http://downloadteens.com">http://downloadteens.com</a>	15
<a href="http://ak.exe.imgfarm.com">http://ak.exe.imgfarm.com</a>	15
<a href="http://travel.cooleimages.com">http://travel.cooleimages.com</a>	14

The top 15 unique malware sample are presented in table 2. Overall, a total of 75 unique malware samples were identified.

Table2: Top 15 malware samples

<i>Malware name identified by ClamAV</i>	<i>Number of samples</i>
JS.Agent-48	533
Trojan.JS-21	128
HTML.Iframe-32	81
HTML.IFrame-10	57
Adware.Trymedia-6	51
JS.Redirect-2	42
Exploit.HTML.Ani	18
Adware.Mywebsearch-5	15
Adware.Casino-22	14
JS.IFRAME-1	14
Exploit.Iframe-1	11
Trojan.Iframe-9	9
JS.Inor-1	9
Trojan.Iframe-3	7
Trojan.Agent-80959	6

Full lists summarizing the results are available under GPL license on the project website.

\*Some domain owners are not aware of malware hosted on their websites. The Honeynet project informed the domain owners of the possible malware hosted on their websites to help them remove it and prevent user infections. False positives are plausible.

## Performance Results:

System was configured to use the maximum available internet bandwidth of 300KB/s on average. A DSL connection was used, which proved to be very unstable for large crawling demands. An average download rate of 350 seconds was identified per 100 MB of content. A separate module was added to the Monkey-Spider system to monitor extractor and scanner statistics. In addition, a stand-alone script has been added to summarize the statistics. A sample report is below.

Table 3: System's extractor and scanner statistics

Total size of archives:	157.8 GB
Average size of an archive:	95.97 MB
### Extractor Statistics ###	
Total time spent for extraction:	1 day, 10:35:35.54
Average time spent extracting an archive:	0:01:13.95
### Scanner Statistics ###	
Total time spent for scanning:	2 days, 4:27:16.07
Average time spent scanning an archive:	0:01:52.14
Average # of malware found per archive:	0.600
Average # of malware found per 100 MB:	0.626

It can be noted from Table 3 that on average an archive of 96MBs was scanned 1.5 times longer than it was extracted. Download rate was identified to be around 350 seconds for 100MB of content and extractor/scanner in total spent approximately 194 seconds for 100MB. This concludes that the analyzer performed almost twice as fast as the crawler, and slept waiting for the new content. Therefore, crawler was the weakest performance component of our system due to limited available Internet bandwidth.

Identifying 0.63 samples of malware per 100MB, allows to conclude that on average, a user browsing the Internet has a 60% probability of receiving at least one sample of malware per 100MB of browsed/downloaded content.

## Advantages vs. Disadvantages of the system setup

### Advantages:

- Properly configured system is able to go through large amounts of content, therefore analyze a broad amount of web-servers
- Results are immediately entered into the database upon scanning and no user interaction is required
- Large anti-virus database provided by ClamAV is capable of identifying a lot of the known malware

### Disadvantages:

- System is unable to identify zero-day malware
- System is unable to identify dynamic malware (For example, malware integrated with JavaScript or Flash)
- Crawling host system is vulnerable to malware infection
- Single antivirus scanner potentially misses some infected content
- No synchronization between the crawler and scanner. This leads to a few potential functionality lacks:
  - First, the crawler is unaware if the web-server hosts any malware. Therefore, it crawls every website in a similar manner. Ideally, a crawler should be able to crawl a domain in more details if malware is found on it.
  - Second, the system is unable to automatically re-crawl previously infected web-servers to monitor changes in malware.

## Conclusion

Honeyclient technology has been around for a number of years to help identify malicious websites. Low interaction honeyclients are fast and easily scalable; however they lack capability of identifying dynamic content or zero-day exploits. On the other side, high interaction honeyclient systems are capable of identifying malicious websites with more accuracy and provide more details, but they are not easy to configure and perform significantly slower. Both low and high interaction honeyclients lack a common framework that will allow synchronizing crawler and analyzer components. Such a framework would extend information gathering about the changes occurring on the malicious URL and its content. Monkey-spider honeyclient can be easily deployed in a short matter of time. Its free licensing and fast analysis capabilities are ideal for gathering a broad picture about the crawled content. However, its functionality is limited and requires a lot of manual work from the administrator. In addition, a single anti-virus serving statically analyzing content potentially misses a lot of malicious URLs.

## References

- Ali İkinci, Thorsten Holz, Felix Freiling. Monkey-Spider: Detecting Malicious Websites with Low-Interaction Honeyclients. Proceedings of Sicherheit 2008, Gesellschaft für Informatik, 2008-04-02.
- ClamAV AntiVirus [[www.clamav.net](http://www.clamav.net)]
- Google AJAX API [<http://code.google.com/apis/ajaxsearch/>]
- Heritrix [<http://crawler.archive.org/>]