INFORMATION SYSTEMS IN MANAGEMENT

Information Systems in Management (2013) Vol. 2 (3) 182−193

MANAGEMENT AND ANALYTICAL SOFTWARE FOR DATA GATHERED FROM HONEYPOT SYSTEM

KRZYSZTOF CABAJ, MAREK DENIS, MICHAŁ BUDA

Institute of Computer Science, Warsaw University of Technology

The paper describes details concerning systems used for analysis and the result of data gathered from two various HoneyPot systems, implemented at Institute of Computer Science. The first system uses data mining techniques for the automatic discovery of interesting patterns in connections directed to the HoneyPot. The second one is responsible for the collection and the initial analysis of attacks dedicated to the Web applications, which nowadays is becoming the most interesting target for cybercriminals. The paper presents results from almost a year of usage, with implemented prototypes, which prove it's practical usefulness. The person performing analysis improves effectiveness by using potentially useful data, which is initially filtered from noise, and automatically generated reports. The usage of data mining techniques allows not only detection of important patterns in rapid manner, but also prevents from overlooking interesting patterns in vast amounts of other irrelevant data.

Keywords: HoneyPot systems, data-mining, monitoring

1. Introduction

Security of computer systems directly connected to the Internet, especially Web applications, becomes more and more important each day. The usage of thousands compromised computers for continuous searching for vulnerabilities in computer systems, inevitably leads to next successful attacks. In order to learn motives, tactics and tools used nowadays by the attackers, HoneyPot systems can

be easily utilized. The HoneyPot is specially crafted and configured machine, or only a chosen service, which is connected to the Internet as a trap for attackers. However, those systems are not used for the production purposes, as its only role is associated with gathering as many information as possible while is being compromised. Software used for implementing various types of HoneyPot systems is easily available. Nonetheless, there is lack of software which could support analysis of gathered data. Using knowledge acquired during many years of HoneyPot system operation and analysis of collected data, the support software was developed and integrated with operational HoneyPot systems.

The paper describes details concerning novel systems used for analysis and the result of data gathered from two various HoneyPot systems, implemented at Institute of Computer Science, Warsaw University of Technology. The first system uses data mining techniques for the automatic discovery of interesting patterns in connections directed to the HoneyPot. The second one is responsible for the collection and the initial analysis of attacks dedicated to the Web applications, which nowadays is becoming the most interesting target for cybercriminals. The paper presents results from almost a year of usage, with implemented prototypes, which prove it's practical usefulness. The person performing analysis improves effectiveness by using potentially useful data, which is initially filtered from noise, and automatically generated reports. The usage of data mining techniques allows not only detection of important patterns in rapid manner but also prevents from overlooking interesting patterns in vast amounts of other irrelevant data.

The paper is organized as follows. The second section describes HoneyPot systems. The third section presents the Miner system, which uses data mining techniques for analysis data gathered from the HoneyPot system. The fourth section is devoted to WebHP system and its monitoring and management software. In section fifth results from initial deployment and operational use of both prototype system are presented. The final sixth, section concludes performed works and indicates future directions and possible improvements.

2. HoneyPot systems

The role of the HoneyPot can be performed by any resource that can be used for observing hostile or unexpected activity. The only common feature of this resource is that it is not used for production purposes. The HoneyPot is mostly specialized machine or software; however, this role can take a fake record in the data base or the account in the important computer system. Any access to the resource, for example, an attempt to read or login, is a sign of unexpected activity. Historically, specially configured computers were used as the HoneyPot system. The configuration enables various monitoring mechanism that during attack gather as many as possible data concerning the attacker activity. For this purpose can be

used logs from operating systems, logs from network devices placed between HoneyPot and Internet or even traces of all traffic directed to it. This solution was ideal for caching and tracking a human attacker but has many drawbacks. The first and the most important is associated with an additional risk. If the attacker detects and disables all monitoring mechanism, the HoneyPot can be used for other hostile activity. Additionally, the initial deployment or cleaning the HoneyPot after a successful attack is very labor intensive. This kind of systems are called high interaction HoneyPots. In the [1] details concerning one of the first well documented development of the HoneyPot and description of further monitoring and tracing real attacker can be found.

In the era of automatic threats, like worms, e-mail viruses or auto-rooters, dedicated high interaction HoneyPots systems used for gathering copies of malicious code new samples are not efficient and very risky. After each infection the HoneyPot system must be cleaned. This process, even with the support of virtualization, is relatively slow. A better solution for gathering information related with malware is usage of low interaction HoneyPots. The low interaction HoneyPot is dedicated software that imitates vulnerable services. Depending on purpose, it can be very simple, for example, only listing for incoming connections and returning standard banners of simulated service. On the other hand, there are very complicated systems dedicated to downloading new samples of malware. This kind of low interaction HoneyPots simulates high level protocols in which vulnerabilities appears, emulates incoming shellcode used by worm during vulnerability exploitation and downloads next stages of the malware. The most important low interaction HoneyPots are HoneyD [2], Nepenthes [3] and its successor Dionaea [4]. During our research on automatic threats, conducted at Institute of Computer Science, only low interaction HoneyPots are used. Due to limitations of available systems, associated with very poor simulation of Web applications, a custom solution was introduced.

3. Miner

The Miner software was developed as a solution that can automatically detect interesting patterns in data gathered from HoneyPot system. It is integrated with low-interaction HoneyPot Dionaea [4], which provides data for later analysis. Using XMPP protocol information concerning all connections from the Internet that reach the HoneyPot are transferred to the separate analytical system and stored in data base. Later, following a cyclical pattern data from last hour, six hours and 24 hours are analyzed using data mining techniques. This process is implemented in Quechua and Quechua-jep modules. Results, detected interesting patterns, are stored in the same data base. Web interface is used for presenting all detected

patterns. For this purpose custom module called miner was developed and integrated with an open source monitoring system carniwwwhore [5].

Figure 1 presents all elements of the system deployed in network of the Institute of Computer Science, Warsaw University of Technology. Presented arrows shows direction of data transfers.

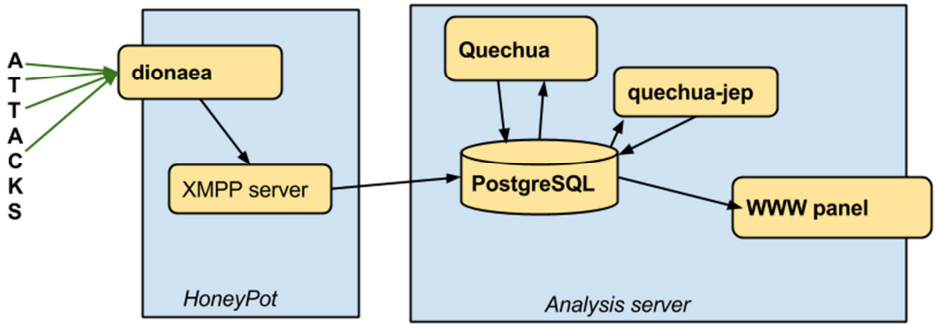

Figure 1. The Miner systems, its Web interface and integration with Dionaea HoneyPot

As previously mentioned, the Miner software uses data mining techniques for analysis. For this purpose two types of patterns are used – frequent sets and jumping emerging patterns. The first pattern was proposed in so called basket analysis, as solution for detection of product sets that are frequently bought together in the markets [6]. In the described system each connection recorded by the HoneyPot is treated as an itemset consisting of five items, associated respectively with source and destination IP address, source and destination port and used protocol. By the definition, frequent set is a subset which appears *minSup* or more times in the analyzed data set. Parameter *minSup* is called minimal support and is given by person who performs analysis. Table 1 presents a sample data set with the connections recorded by HoneyPot.

Table 1. Sample data set used in described example

	Protocol	Source IP	Source Port	Destination IP	Destination Port
1	tcp	10.1.XX.XX	54333	192.168.YY.YY	80
2	tcp	10.1.XX.XX	54333	192.168.YY.YY	80
3	tcp	10.1.XX.XX	54333	192.168.YY.YY	80
4	tcp	172.16.ZZ.ZZ	42356	192.168.YY.YY	80
5	tcp	172.16.ZZ.ZZ	42456	192.168.YY.YY	8080
6	tcp	172.16.ZZ.ZZ	44895	192.168.YY.YY	1080

Assumed that we set parameter *minSup* to the value three, various frequent sets can be detected, for example, <tcp, *, *, *, * >, <tcp, *, *, *, 80>, <tcp, *, *, 192.168.YY.YY, 80>, <tcp, 10.1.XX.XX, 54333, 192.168.YY.YY, 80> or <tcp, 172.16.ZZ.ZZ, *, 192.168.YY.YY, *>. Asterisk sign presented in the example frequent sets respectively supports initial item sets in ranges, 1-6, 1-4, 1-4, 1-3 and 4-6. The most interesting are the last two which are called maximal, due to the fact, that there is no other detected frequent sets in this data set that are over-sets of them. For further analysis only maximal frequent sets are considered. They are searched in all patterns which are discovered by Miner software using Apriori algorithm.

The second pattern, used in the developed system is called jumping emerging pattern (JEP) [7]. This kind of pattern could be defined between two data sets in which frequent sets are detected. The JEP is a frequent set that is detected in one data set and is not present in the second one. In the Miner system frequent sets are detected in the cyclic pattern in various length intervals: one hour, six hours and 24 hours. JEPs are detected between two adjacent intervals, that have the same duration In case that some repeated activity interacts with the HoneyPot for longer period, frequent set associated with this events due to usage of JEP is presented only once, in the first interval. The usage of JEPs highlights changes in detected frequents sets, reduces number of patters that should be inspected by human operator and in the effect reduce possibility of important pattern omission. Figure 2 presents Web interface of the Miner with list of performed detections of frequent sets in variable length intervals.

Miner - Operations

id	from	to	interval	No. of all frequent itemsets	No. of non generating frequent itemsets	No. of JEPs	No. of JEPs with remote values	Tag
31616	27/09/2013 16:00:00	27/09/2013 17:00:00	1 hours	3	1	0	0	cron-test
31615	27/09/2013 15:00:00	27/09/2013 16:00:00	1 hours	3	1	1	0	cron-test
31614	27/09/2013 14:00:00	27/09/2013 15:00:00	1 hours	1	1	1	0	cron-test
31613	27/09/2013 13:00:00	27/09/2013 14:00:00	1 hours	0	0	0	0	cron-test
31612	27/09/2013 12:00:00	27/09/2013 13:00:00	1 hours	3	1	1	0	cron-test
31611	27/09/2013 06:00:00	27/09/2013 12:00:00	6 hours	13	2	1	0	cron-test
31610	27/09/2013 11:00:00	27/09/2013 12:00:00	1 hours	0	0	0	0	cron-test
31609	27/09/2013 10:00:00	27/09/2013 11:00:00	1 hours	0	0	0	0	cron-test
31608	27/09/2013 09:00:00	27/09/2013 10:00:00	1 hours	0	0	0	0	cron-test
31607	27/09/2013 08:00:00	27/09/2013 09:00:00	1 hours	7	1	1	1	cron-test
31606	27/09/2013 07:00:00	27/09/2013 08:00:00	1 hours	3	1	1	0	cron-test
31605	27/09/2013 06:00:00	27/09/2013 07:00:00	1 hours	7	1	1	0	cron-test
31604	27/09/2013 00:00:00	27/09/2013 06:00:00	6 hours	9	2	2	0	cron-test
31603	27/09/2013 05:00:00	27/09/2013 06:00:00	1 hours	0	0	0	0	cron-test
31602	27/09/2013 04:00:00	27/09/2013 05:00:00	1 hours	0	0	0	0	cron-test
31601	27/09/2013 03:00:00	27/09/2013 04:00:00	1 hours	0	0	0	0	cron-test
31600	27/09/2013 02:00:00	27/09/2013 03:00:00	1 hours	0	0	0	0	cron-test
31599	27/09/2013 01:00:00	27/09/2013 02:00:00	1 hours	0	0	0	0	cron-test
31598	27/09/2013 00:00:00	27/09/2013 01:00:00	1 hours	0	0	0	0	cron-test
31597	26/09/2013 00:00:00	27/09/2013 00:00:00	1 days	21	4	2	0	cron-test
31596	26/09/2013 18:00:00	27/09/2013 00:00:00	6 hours	7	1	1	1	cron-test
31595	26/09/2013 23:00:00	27/09/2013 00:00:00	1 hours	15	1	1	1	cron-test

Figure 2. Appearance of the Miner system Web interface, with list of performed pattern discovery in variable length intervals

In the table various detailed information concerning detection of frequent sets and JEPs are presented. In subsequent columns id of given calculations, start and stop time of the interval, interval length, number of detected frequent sets, number of maximal frequent sets, number of detected JEPs and number of interesting patters are presented. Figure 3 presents details concerning detected by the Miner system frequent sets and JEPs.

Miner - Operation

Operation **31554** **TAG:** cron-test
From: **Sept. 25, 2013, noon** to **Sept. 25, 2013, 1 p.m.**
Duration: **1 hours** (3600 seconds)

id	counter	proto	remote host	remote port	local host	local port	JEP	Interesting
553680	5	tcp			127.0.0.1	3389	YES	no
553681	9	tcp	111.253.250.137		127.0.0.1		YES	**YES**

Figure 3. Appearance of the Miner system Web interface with details concerning detected frequent sets in given interval

More results concerning information that can be detected using the Miner system are discussed with details in the section number five.

4. WebHP and HPMS software

WebHP and HPMS (HoneyPot Management System) software was developed due to limited capabilities associated with gathering details connected with data exchange in application layer between attacker and a low interaction HoneyPot. WebHP was developed as specialized data capture script implemented in PHP language. It must be placed in each monitored page of prepared Web HoneyPot static pages or an application. It is responsible for logging all request send from attacker to data base used by HPMS management system. Additionally, in the implemented Web HoneyPot custom error page was prepared, which included data logging script, too. This allows the capture of any request, even if requested page is not present in the Web HoneyPot. The HPMS system was implemented in Python language using Django framework. It allows easy access to all data captured by Web HoneyPot, for example, searching for interesting requests and plotting activity in given time range. Moreover, the user can define rules, which automatically tag all requests matching certain conditions. Figure 4 presents elements of WebHP with HMPS system deployed in network of Institute of Computer Science, Warsaw University of Technology. Figure 5 shows sample screen shot of the HPMS Web interface.

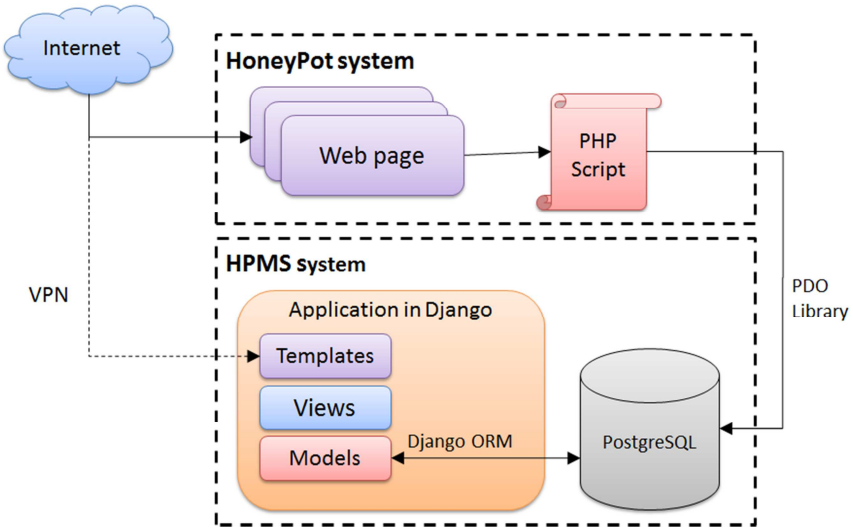

Figure 4. Integration of WebHP and HPMS systems

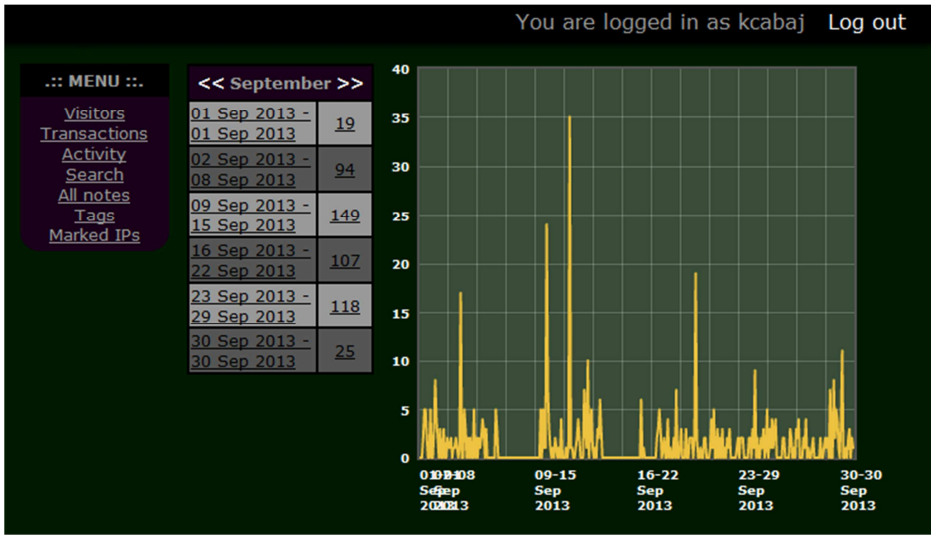

Figure 5. Appearance of sample HPMS Web interface

5. Results

Both systems described in the previous sections were deployed at the end of the year 2012 in the network of Institute of Computer Science, Warsaw University of Technology. The HoneyPot sensors are placed in the same network using IPv4 addresses, few addresses in distance one from another. Both sensors are freely available from the Internet. The access to the management interfaces were secured only for users working internally or those who have valid access to the internal network via VPN.

Even though HoneyPot sensors are not used for any other activity, and its addresses were not specially announced, during this period of time vast amounts of data have been captured. The Dionaea HoneyPot, which was integrated with the Miner system, received from September 2012 to the end of September 2013, more than 827,5 thousands connections. The WebHP, which was analyzing only connections directed to the WWW services, received from the beginning of November 2012 to the end of September 2013 more than 22,7 thousands connections. This numbers proves that analysis of gathered data manually without specialized software is almost impossible. In the following part of this section the most interesting findings, discovered using implemented management systems, are presented.

The first observation concerning data gathered by both systems shows that automatic scanning is performed for many addresses in given network, one by another. In most cases when some activity from the suspected address was observed in WebHP system, even broader data ware captured by Dionaea integrated with Miner software. Figure 6 shows exemplary request logged by WebHP that checks if it can be used as open proxy. Attacker uses IP address 115.24.164.179 and connects to the HoneyPot at 26 September 23:57. In the similar time, Miner software in six hour interval from 18:00 to 0:00 at 26th September detects frequent set which have item corresponding to this same IP address. Figure 7 shows detected pattern in the user Web interface of the Miner system.

Last 24 hours | Last 48 hours | Last 96 hours | Last 192 hours | Sl

Transaction	Visitor IP	Date/Time	REQUEST_URI	Tags
22577 G [0]	157.56.229.190 [0]	26 September 2013 15:09:12	/robots.txt	M$ BingBot
22578 G [0]	157.56.229.190 [0]	26 September 2013 15:10:10	/	M$ BingBot
22579 H [0]	110.249.212.236 [0]	26 September 2013 18:27:55	/	
22580 G [0]	157.55.32.189 [0]	26 September 2013 23:54:32	/robots.txt	M$ BingBot
22581 G [0]	157.55.32.189 [0]	26 September 2013 23:55:10	/	M$ BingBot
22582 G [0]	115.24.164.179 [0]	26 September 2013 23:57:42	http://www.google.com.hk/	PROXY

Figure 6. Searching for proxy logged by WebHP presented in HPMS Web interface. Marked line from IP address 115.24.164.179

Detected frequent set has support equal to16. This is caused by fact that this scanner searches proxy in various ports, not only at the standard port 80. In this case are checked, for example, port number 8888, 808, 8080, 3128, 8118 and 1080.

Miner - Operation

Operation **31596** **TAG:** cron-test
From: **Sept. 26, 2013, 6 p.m.** to **Sept. 27, 2013, midnight**
Duration: **6 hours** (21600 seconds)

id	counter	proto	remote host	remote port	local host	local port	JEP	Interesting
553755	16	tcp	115.24.164.179		127.0.0.1		YES	YES

Figure 7. Detected by the Miner software pattern, which represents searching for proxy performed from IP address 115.24.164.179

The main advantage of the Miner system is associated with patterns discovery. In initial assumptions each detected pattern represents logged activity, which should be manually inspected by the system operator. As the expected number of detected patterns should be smaller than the number of logged events. During the initial deployment phase, when real data gathered by Honeypot were analyzed, some additional constraints are introduced. In effect patterns that do not carry interesting knowledge are omitted. For this purpose in subsequent steps of system development maximal frequent sets, jumping emerging patterns and interesting patterns are proposed. Maximal patterns cover from the operator all detected by subsets. When the maximal pattern, for example, <tcp, 10.0.XX.XX, *, 192.168.YY.YY, 80> is discovered in analyzed data, additionally its subsets are detected, too. In effect an operator has to search useful data in many other frequent sets, for example, <tcp, *, *, *, *>, <tcp, *, *, *,80>, <*,*,*,192.168.YY,80>. The second improvement reduce additional data when hostile activity is performed for longer period. If data from HoneyPot is analyzed only using discovery of frequent sets, than longer hostile activity produce many very similar or even identical patterns. The usage of pattern called JEJ reduces number of generated patterns only to situations in which something changes in the analyzed data. In the effect the first pattern will be generated, when hostile activity starts and the second when its stops. The last improvement is associated with frequent sets discovery behavior, that produces events which carried little new knowledge. The used algorithm tried to generate any frequent sets. In the effect, when first version of the Miner system was used in intervals with little activity, completely useless patterns were detected, for example, <tcp, *, *, 192.168.YY.YY, *> which represents connections using tcp protocol to our HoneyPot with any source address or port. Due to this fact, the definition of interesting patterns was introduced. The interesting pattern is such

frequent set, that is JEP and contains items associated with source port or address. All described in this section improvements reduce number of patterns, that the operator must check. In the analyzed period for more than 827,5 thousands events almost 67 thousands of frequent sets are discovered. In this number there are about 11 thousands of maximal frequent sets, about 5 thousands of JEPs and about 2 thousands interesting frequent sets. These numbers show a reduction of events that the operator must analyze. Moreover, when the operator does not have to find interesting events in vast amount of useless data some interesting data, firstly omitted can be observed. Figure 8 shows a sample analysis concerning one week time frame prepared by the Miner software.

id	counter	proto	remote host	remote port	local host	local port	JEP	Interesting
554156	40	tcp			127.0.0.1	23	no	no
554157	24	tcp			127.0.0.1	25	no	no
554170	23	tcp			127.0.0.1	4899	no	no
554171	11	tcp			127.0.0.1	5900	no	no
554179	12	tcp	123.151.42.61		127.0.0.1		no	no
554225	16	tcp		6000	127.0.0.1	3128	no	no
554227	8	tcp		4935	127.0.0.1	3389	no	no
554230	11	udp		12051	127.0.0.1	5060	no	no
554231	12	udp		12052	127.0.0.1	5060	no	no
554232	7	udp		12053	127.0.0.1	5060	no	no
554234	8	udp	188.138.33.215		127.0.0.1	5060	no	no
554221	6	tcp		5004	127.0.0.1	1433	YES	YES
554223	6	tcp	219.235.8.245		127.0.0.1	1433	YES	YES
554229	5	tcp	188.116.21.86	42343	127.0.0.1		YES	YES
554226	11	tcp		6000	127.0.0.1	3306	YES	YES
554158	22	tcp			127.0.0.1	80	YES	no
554233	7	udp		12054	127.0.0.1	5060	YES	YES
554236	8	udp	50.30.37.9		127.0.0.1	5060	YES	YES
554237	6	udp	188.138.41.34		127.0.0.1	5060	YES	YES
554228	5	tcp		6000	127.0.0.1	8080	YES	YES
554222	6	tcp	122.0.66.98		127.0.0.1	1433	YES	YES
554235	5	udp	173.242.117.164		127.0.0.1	5060	YES	YES
554224	6	tcp	119.100.21.141		127.0.0.1	1433	YES	YES
554220	5	tcp	218.7.37.194		127.0.0.1	22	YES	YES

Figure 8. Patterns detected by the Miner system in week interval. Very interesting patterns representing scanning activity from fixed source port (6000, 4935, 12051, 12052, etc.)

It is interesting that there are some frequent sets representing scanning activity performed from various IP addresses, which use the same source port. The activity of vulnerability scanners that use source port 6000 is well known in security field [8]. However , the detection of scanners that use source port 4935, 12051, 12052, 12053 and 12054 was astonishing. Moreover, without an automatic detection of patterns and implemented filtering function those facts cannot be revealed.

The main advantage of the WebHP is associated with gathering of application data, which can give better insight into attackers intentions. The automatic tagging feature can save an operator time and give opportunity to analyze only unknown activity. During search with the prototype of the system almost twenty distinct tags were discovered and configured in the HPMS systems. Some of them (Proxy and MS BingBot) can be observed in the Figure 6. Even more interesting results can be acquired when the ability to interact with the attacker is used. During conducted experiments, in the WebHP guest book without any "human detection" mechanism was deployed. After few months of inactivity, well organized process of posting hostile links began. During only one week more than 10 thousands of links were added. In this attack 480 distinct IP addresses were used. Further analysis shows that there were two kinds of Bots. The first, which was observed in 388 machines, sequentially placed new posts to the guest book. The second, smaller group which contains 92 machines only checks if guest book is still available, and posts are successfully added. Figure 9 presents plot from HPMS system showing an hourly number of distinct access to guest book pages. Before attack the average of 3 to 5 events occurred in one hour. However, during the attack more than one hundred request are sent to the HoneyPot. The attack has been stopped administratively by disabling the guest book.

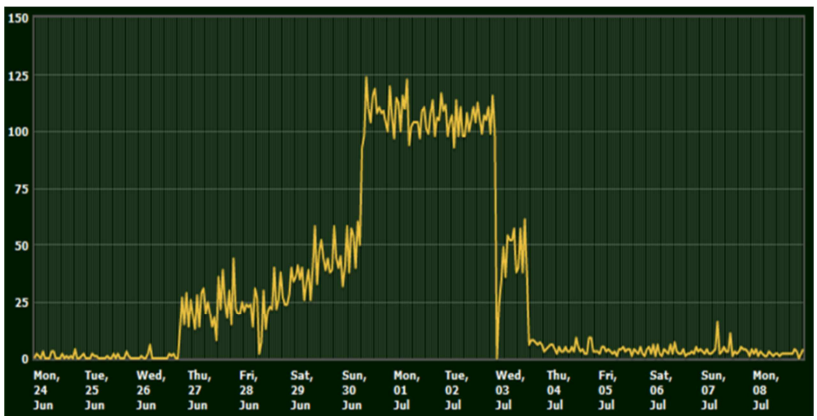

Figure 9. Plot from HPMS showing activity during SPAM attack at the HoneyPot guest book

6. Conclusions

Both implemented systems, the Miner and WebHP with HMPS software, were developed using experience from operation use of various HoneyPots in Institute of Computer Science network. The first described system uses data mining techniques for analysis of data gathered from Dionaea HoneyPot. The usage of JEPs and frequent sets indicates the most important data for analysis by human operator. Additionally, filtering achieved by usage of those patterns allows discovery previously unknown patterns, for example, the behavior of some scanning programs, that use hardcoded source ports.

The second system gives insight into data transmitted during attacks on Web sites and Web applications with the level of details that previously was not able to be achieved. Automatic application of tags saves an operator time and allows to analyze only new, previously unseen activities. The ability of the better interaction with attackers gives additional data for further analysis, for example, potentially hostile links placed at the guest book.

Almost year of the operational usage of both systems proves that both systems increase knowledge about attacks directed to the implemented HoneyPots. Functionality build into the systems that does some tedious work, automatically increases productivity of operators and reduces the possibility of interesting events omission.

REFERENCES

[1] Cheswick B. (1992) An Evening with Berferd in which a cracker is Lured, Endured, and Studied, In Proc. Winter USENIX Conference

[2] Provos N., Holz T. (2008) Virtual Honeypots: From Botnet Tracking to Intrusion Detection, Addison-Wesley

[3] Baecher P., Koetter M., Dornseif M., Freiling F. (2006), The nepenthes platform: An efficient approach to collect malware, In Proceedings of the 9 th International Symposium on Recent Advances in Intrusion Detection (RAID06)

[4] dionaea catches bugs, http://dionaea.carnivore.it/ [25.11.2013]

[5] Carniwwwhore , http://carnivore.it/2010/11/27/carniwwwhore [25.11.2013]

[6] Agrawal R., Imielinski T., Swami A. (1993) Mining Association Rules Between Sets of Items in Large Databases, Proceedings of ACM SIGMOD Int. Conf. Management of Data,

[7] Dong G., Li. J. (1999) Efficient mining of Emerging Patterns: Discovering Trends and Differences. In Proceedings of the Fifth International Conference on Knowledge Discovery and Data Mining, San Diego, USA (SIGKDD'99), 43–52

[8] White G.N. (2010) What's Up With All The Port Scanning Using TCP/6000 As A Source Port?, https://secure.dshield.org/diary/What%27s+Up+With+All+The+Port +Scanning +Using+TCP6000+As+A+Source+Port%3F/7924 [25.11.2013]