LOBSTER Deliverable 3.4

Integrating a LOBSTER Sensor with Global Threat Intelligence

Final Report

Table of contents

Acknowledgements

The work presented in this document has been conducted in the context of the EU Framework Programme project LOBSTER. LOBSTER is funded by the European Commission as well as by the industrial partners. Their support is appreciated.

1. Overview

Threat Collection and Threat Analysis are critical to collecting samples of malware and malware variants, generating signatures for such malware, and tracking the growth and prevalence of such threats online. Threat Collection and Threat Analysis are critical to tracking threats effectively, and critical to effectively generating signatures for critical new malware. As signatures are generated very quickly from samples, rapid collection of samples dominates overall signature generation time. For this reason, accelerating sample collection is critical to providing increasingly better signature-based defenses. Additionally, large organizations need insight into prevalence and growth of threats online. Fortunately, the very same threat collection infrastructure used for collection of malware samples can also be used for collection of malware statistics helping better and more intelligently inform people of the state of online threats on a global scale, hence the phrase, “Global Threat Intelligence.”

Malware of a *metamorphic* and *polymorphic* nature sometimes require signature providers to generate multiple signatures for the same sample, or even collect multiple samples as the malware changes itself over time. These attacks are on the rise. For this reason, they warrant investigation of alternative approaches for detection such threats and collecting samples of such malware. For these reasons, LOBSTER has been coordinating deployment and evolution of the NEMU sensor developed and maintained by the FORTH institute in Crete. NEMU has the ability to detect polymorphic code within network traffic. If NEMU has the ability to catch polymorphic attacks that would evade a reference engine, then NEMU could be very useful in accelerating threat collection and improving detection of very dangerous polymorphic malware.

For these reasons, Task 3.4 “Integration of a LOBSTER sensor with the DeepSight framework” focused on integration of NEMU with a Reference Engine Filter (REF) that is part of the the DeepSight framework. If NEMU were detecting a sufficient number of malware variants more quickly than the baseiine components of the Reference Engine, then NEMU could be a very valuable extension to threat collection infrastructures powering Global Threat Intelligence and other commercial services.

Our analysis revealed that NEMU does in fact occasionally detect a variant more quickly than some of the baseline components. However, such cases were rare, occurring less than once per week among thousands of alerts.

2. Summary of Task 3.4

Work Package 3.4 states, “Given the potential value of LOBSTER alerts, and given the potential value of feedback to the LOBSTER system … this task could use LOBSTER alerts to cue … sensor framework and adjust defenses... The end result of this task is to provide a report both describing the most effective gains from LOBSTER alerts, and, in so far as it is possible to share results within legal and contractual privacy constraints, and the constraints of protecting absolute confidentiality of (proprietary) Preexisting Know-How, describing information and data observed by (Global Threat Intelligence systems and other commercial services) that might be helpful to LOBSTER in monitoring internet health on European scale.”

3. NEMU – a Polymorphic Shellcode Detector

NEMU is a prototype developed by FORTH Institute that uses CPU emulation in the detection of polymorphic shellcode at the network level. A worm that employs polymorphic shellcode to commence and propagate its outbreak, typically consists of some payload shellcode that is encrypted differently for each attack and is pre-pended with its decryption routine to enable it to self-decrypt. Because the decryption routine must remain constant, signatures are often written that detect the corresponding patterns, however this can only occur after an attack has being recognised in the first place. The NEMU approach is entirely different, it assumes that the code under examination is an executable and runs it using a CPU emulator. Various heuristics are then used to determine if polymorphic behaviour is evident, e.g. if the code actually runs a series if instructions without crashing and if the memory space used by the payload is being accessed etc. NEMU's approach uses passive monitoring and requires no signatures and by definition has the ability to discover zero day polymorphic worm attacks. NEMU is a prototype based on a paper by the software's authors entitled "Network-Level Polymorphic Shellcode Detection using Emulation".

4. Reference Engine

Evaluation of NEMU is against the capabilities of a reference engine. We should note that we do not specify the reference engine as, for liability reasons, we do not want to speak ill of competing tools, or identify specific gaps in detection capabilities of any tools. This reference engine is available as a command line tool and, among other things, it has the ability to scan tcpdump formatted packet files and output information on attacks and anomalies such as protocol irregularities.

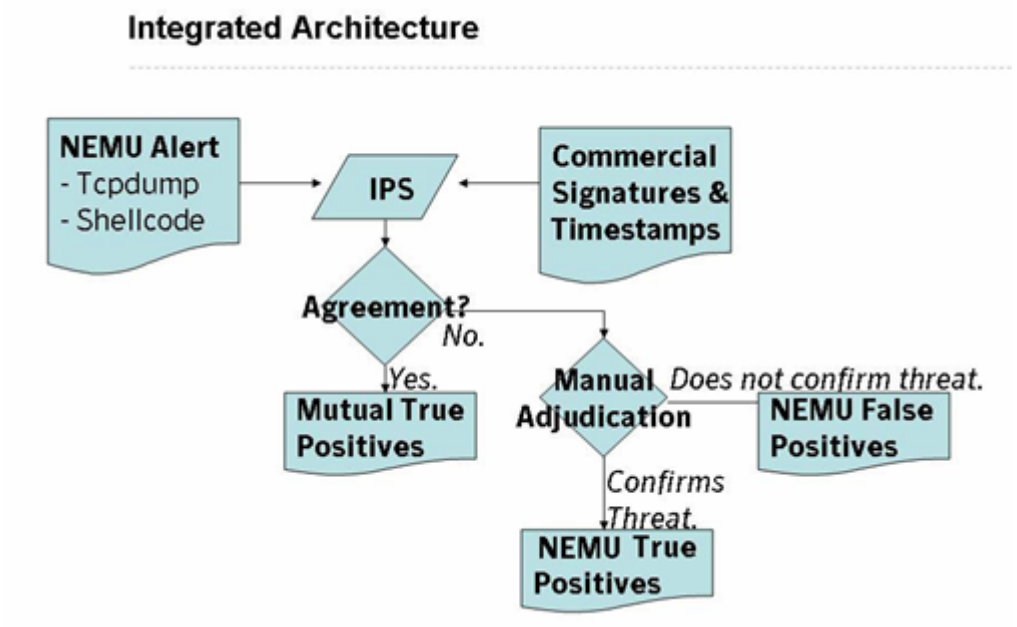

Figure 1. Integrated Architecture for the Experiment Plan

The architecture for integration is depicted above. The Reference Engine is labelled "IPS" for Intrusion Prevention System in the diagram above because, outside of this context, it is often configured to operate as an Intrusion Prevention System. We note the reference engine has a common interface to Global Threat Intelligence systems.

5. Experiment Plan

As described in the Overview section, a potential commercial value of NEMU exists in potential advantages of NEMU's non-signature based detection techniques over signature based detection techniques. For this reason, we constructed an experiment to measure this value. The goal of the experiment is determine to what extent, if any, NEMU detects malicious behaviour in network traces that evade detection by a reference engine. Where that happens, we determine whether or not the behaviour missed by the reference engine is a type of behaviour to be concerned about. For this reason, adjudication requires manually inspecting alerts that evade REF IPS.

NEMU alerts

As part of the LOBSTER project a sensor was deployed by FORTH that monitors traffic on their network. This is no more than a computer running NEMU which is passively listening to the network interface. Alerts arising from such monitoring are uploaded daily for us to download via secure HTTP. Corresponding to each alert are 4 files; a pcap trace of the traffic that contains the alert, a raw TCP stream of the traffic that contains the alert, a human readable alert file and a file containing the decrypted shellcode contained in the trace.

Reference Engine Filter (REF) Intrusion Prevention System (IPS)

The pcap trace files are provided as input to the reference engine. For the duration of this experiment two version of the reference engine were available: versions 9 (released on Nov. 30th 2006) and version 12 (released on Feb. 12th 2007).

Signatures

Because we are contrasting signature based to non-signature based systems, it is important to note the range of dates from which signatures were used. Signature sets dated 8th Jan 2007 to 21st Feb 2007 were used. In each comparison, the signature set used in the experiment preceded the date of the NEMU alerts.

Agreement

Regarding a particular alert trace, agreement is defined to exist between NEMU and the reference engine when the REF IPS attributes a reference engine event of "attack" to the alert. Initially we assume that alerts that do not trigger valid REF IPS attacks are in fact valid attacks simply evading REF

IPS, then we confirm or disprove that assumption. Also, we pursue a similar manual adjudication process whenever a NEMU alert only triggers a REF IPS "anomaly" event.

Manual Adjudication

This is required when an alert evades the reference engine and expertise is required to, at a minimum, closely examine the files relating to the alert to determine if it is indeed of a malicious nature. Assistance was provided by experts inside and outside of Europe for this portion of the experiment, but costs of such analysis outside the European Union were not and will not be billed to the LOBSTER effort.

6. Experiment Execution

The Alert Traffic Traces

The traffic traces for the experiment comprise 43 days of alerts, dated 11th Jan 07 to 22nd Feb 07, detected at the NEMU sensor deployment on FORTH's network in Crete. This amounts to 2221 individual alerts. Identical decrypted shellcode is typically common to many alerts to the extent that 152 distinct decrypted shellcodes were encountered. Alerts that contain the same decrypted shellcode we will define as belonging to the same *alert class*. The most commonly targeted ports in these alerts are 445 and 2967, but attacks against ports 1051, 80, 1025, 42 and 135 are also present.

Reference Engine Version 9 Scan.

Reference Engine Version 9, using a signature set dated 8th Jan 07, was used to process the 2221 .pcap alert files. A total of 2204 of these alerts were detected immediately by the reference engine's attack signatures.

Signature Label	Trigger Count
HTTP BO	7
HTTP Content Smuggling	5
MSDTC BO	16
ASN1 BO	98
PNP BO	36
RPC_DCOM_Attack	17
WINS BO	2
LSASS WORM	566
RPC NETAPI32 BO	857
SMB BO	5
Other BO	595

Table 1. Signatures Catching Alerts.

It is often the case that a single NEMU alert may cause more than one attack signature to trigger. We regard the signature associated with the first valid attack event listed in the reference engine output, of an alert scan, as the one attributed to catching the alert.

The data in the Table 1 therefore captures the profile with which different signatures are attributed to catching alerts as opposed to the number of times with which they actually trigger.

The creation date of the above signatures, range from 3rd Dec 2003 to 6th Feb 2006. No alert caused only reference engine anomalies to trigger.

Alert Class ID	Evading	Caught
032d194ab129c18a24eb4cf0dda8d52e	2	131
3d058cc5c35e7337cd9f713ef0ee70fa	2	111
4ace5fca76f1b4c7c426f34a5fd2523f	2	38
5294380c4047f09c052d96c6e6111eda	2	626
7d62e435aef9747e0ff777c27d7a9f01	3	580
9e8bcd1857d8aa59c1ac589743bcd983	1	139
a5567b03108696a6b0d77f55d024e7c2	1	3
a8c58a8ca99c6fa6b73a53d00bf7ed0e	1	0
f24f8ebae155c085b49e4888e43bb1ed	2	92
f681ceecc333c4c78f5982c3dd8a9b48	1	0

Table 2. Alert Classes Evading Reference Engine v.9.

Table 2 lists the 17 alerts that evaded REF IPS Version 9. These are categorised by 10 alert classes with each alert class represented by the md5 hash of the decrypted shellcode. Reflecting the polymorphic nature of the alerts, all but 2 of the alert classes had other instances that were caught by REF IPS. Since only 17 of 2221 alerts were missed by REF IPS, we can say that REF IPS detected 99.23% of the attacks detected by NEMU, and that REF IPS missed less than 1% of the attacks detected by NEMU. It should be noted that REF IPS also detects a broad range of non-polymorphic threats, but those strengths are not factored into this experiment.

The Two alert instances that evaded the reference engine were sent to experts for manual adjudication. These were as follows:

1) Alert 20070119_10:17:56.551395_86.69.13.65 containing shellcode ID 9e8bcd1857d8aa59c1ac589743bcd983.

2) Alert 20070119_15:13:34.097617_83.167.152.4 containing 032d194ab129c18a24eb4cf0dda8d52e.

The first was deemed to be an actual attack and therefore the first NEMU true positive of the experiment that evaded the reference engine. On analyzing the other alert(s) as described below, it turned out to be the only true positve from this phase of the experiment. In context of 151 alert classes detected by both NEMU and REF IPS, the result is that this single NEMU sensor might improve REF IPS signature set quality by nearly 1%, specifically 00.67% where it is possible to overcome the privacy constraints of applying such sensors to customer or service provider networks. It was not clear initially from this experiment whether additional NEMU sensors could provide near linear additive value, or whether the point of diminishing returns would be

reached quickly. However, from applying NEMU to a number of other traces from non-disclosed sources, the gain to be had from NEMU does not appear to outweigh the challenges of privacy constraints for large scale deployment.

The pcap file of the 2nd alert was deemed to be ill-formed for reference engine input. The pcap file was missing a handshake and was therefore stateless. It was learned that REF IPS version 9 must operate on state-full pcap files and reference engine version 9 did not perform the *sanitization* required in order that stateless pcaps can be treated as state-full. Reference engine version 12, which was near release at the time, provides this functionality, as do commercially available tools.

The reference engine version 9 scan was re-run with the *updated signature* set of 21st Feb 07. These updates did not alter the numbers of NEMU alerts evading the reference engine version 9 in any way. This is primarily because of the nature of 1) 10:17 and because the flaw in REF IPS performance on 2) 15:13 was not a flaw in the signature set but rather a flaw in overall resilience to other evasion techniques requiring sanitization.

Reference Engine Version 12 Scan

The main difference with the reference engine version 12 scan relates to its ability to perform sanitisation of mal-formed TCP handshakes in the pcap files before conducting analysis of the TCP packets. Again the earlier signature set from 8th Jan 07, was used in the first run. After running the scan on the complete batch of alerts, the number of alerts evading the reference engine was reduced from 17 to 7. The 7 alerts that evaded the latest reference engine at the time relate to 5 alert classes, as seen below

Alert Class ID	Evading	Caught
032d194ab129c18a24eb4cf0dda8d52e	2	321
4ace5fca76f1b4c7c426f34a5fd2523f	1	39
7d62e435aef9747e0ff777c27d7a9f01	2	581
9e8bcd1857d8aa59c1ac589743bcd983	1	139
a8c58a8ca99c6fa6b73a53d00bf7ed0e	1	0

Table 3. Alert Classes Evading Reference Engine v.12.

The alert file 20070119_10:17:56.551395_86.69.13.65 that contained the NEMU true positive proved insensitive to the sanitisation issue. However, we note as above that this would be less than a 1% improvement in detection of attack classes. Also, this attack may be detectable through tools other than line-wire IPS. Last, we note that the other improvements in REF IPS raised the rate of detection from 99.23% (2204/2221) to 99.68% (2214/2221). From the remaining alert instances, the following was observed;

- The two alerts evading the reference engine that belong to alert class 7d62e435aef9747e0ff777c27d7a9f01, namely 20070129_12:18:31.028575_81.186.53.163 and 20070129_14:01:13.727288_81.186.53.246, were detected as having invalid dumps and therefore could not be sanitised. The problem is that the pcap file only contains part of the attack and it was noted that “sniffer started late” errors were seen. We should note that if a

production IPS had been running from before the attack started, the attack would likely have been detected and prevented. We also note the pcap files relating to these two alerts are much smaller than is usually the case with these alert files. (These files contain only 441 and 1782 bytes rather than 6000/7000 bytes contained in complete pcaps). For these reasons, since REF IPS was not given a proper PCAP trace, we will not regard these attack instances as evading the reference engine. The fact that 581 sibling instances were caught by REF IPS also supports this.

- The alert 20070122_23:10:06.263852_84.193.169.146, contained shellcode referenced by a8c58a8ca99c6fa6b73a53d00bf7ed0e, evaded the reference engine and this finding instigated development of a new signature. We are inclined to count this as the second NEMU TP evading the reference engine, and the first NEMU TP for which line-wire signature generation is very appropriate.

- The members of the alert class 032d194ab129c18a24eb4cf0dda8d52e include; 20070119_15:13:34.097617_83.167.152.4 and 20070204_00:27:35.376761_81.158.164.183, both were causing invalid attack types raising events such as “TCP unexpected timestamps” and “too many TCP retransmitted segments.” In this case, the sanitization process was getting the client to server direction wrong in the case of these 2 dump files. Setting a reference engine configuration parameter correctly would trigger the attack signatures when their sanitised pcaps were rescanned. This proved to be the case, therefore these 2 alerts were not considered as evading the reference engine, but we should be conscious that a non default configuration was required.

- Adjudicating alert instance 20070215_10:37:38.888695_81.182.230.53 from class 4ace5fca76f1b4c7c426f34a5fd2523f proved ambiguous. However, even if we consider it a TP in favor of NEMU, this would only be a third alert instance of 2,221 instances where NEMU detected something that a production IPS would seem likely to miss. 3 of 2221 is much less than a 1% improvement performance. Specifically, it is roughly a 00.1% improvement in performance of a system already beating 99.86% detection rates of polymorphic threats, and performing yet better on non-polymorphic threats. If we strictly consider attack classes, not weighted by prevalence of attacks, this could be an improvement of 3 out of 152 attack classes, or roughly a 1.97% improvement in the range of attack classes detected. This could be a substantive improvement if the privacy constraints can be overcome for modest scale deployment to customer or service provider networks. However, it might be possible to achieve such improvement more cost effectively through growth of conventional threat collection infrastructure.

The reference engine version 12 scan was re-run with the *updated signatures* sets dated 21st Feb 07. However, as before, the signature updates did not alter the numbers of NEMU alerts evading the reference engine version 12 in any substantive way.

Reference Engine Version 9 versus Version 12 Summary

Comparing the results of the scans produced from the 2 different versions of the reference engine is effective in determining how critical the sanitisation issue is to such signature based detection technologies. Sanitisation was verified to be the sole reason for the difference between performance of version 9 and version 12. This was confirmed when the sanitized pcaps were scanned by reference engine version 9.

Reference Engine Version	Alert Count	Alerts Caught	Alerts Evading	Alerts Evading (sans traces incomplete)
9	2221	2204	17	15
12	2221	2214	7	3

Table 4. Effect of Sanitisation on Reference Engine evasion

We see in Table 4 above that improving sanitization resulted in improvements of 0.4% (10/2221) whereas integration of a NEMU sensor could only improve performance by 0.1% (3/2221).

Table 5, below, is a summary of the experiment. The revised alert evasion numbers account for invalid pcap files (pcap files that did not capture the complete trace) and correcting reference engine configuration settings. The ability of reference engine version 12 to sanitize pcap files (in proper configuration) meant 12 alerts were prevented from evading the reference engine, leaving only three alerts evading the reference engine from over two thousand alerts orginally. The sanitization issue aside we can state we had 2 NEMU true positives that evaded the reference engine, and that 1 other alert remains ambiguous.

Alert Count	Mutual TP's	NEMU TP's	NEMU FP's	Incomplete Traces	Ambiguous
2221	2216	2	0	2	1

Table 5. NEMU statistics

It should be noted that since it was not possible to deploy the reference engine to unfiltered traffic at the NEMU sensor point, it is not possible to identify any false negatives for NEMU, and the reference engine would have likely detected non-polymorphic attacks in addition to the polymorphic attacks detected, as designed.

Last, we should note our reasons for considering NEMU primarily for threat collection and not for customer deployment. Specifically, in analyzing NEMU performance with a multi-GHz processor and a GB of RAM on attack rich datasets, NEMU managed throughput of under 2 Mbit/s. Although operation of NEMU on benign traffic sets can handle 60 Mbit/s, this is still substantially below the GB/second rates that are currently commercially competitive. However, despite current focus of the market on line-speeds, from a scientific perspective, detecting such large volumes of attacks through non-signature based techniques without false positive is a very impressive accomplishment.

7. Conclusions

1. NEMU is a useful prototype capable of detecting thousands of attacks without false positive.

2. A total of 3 NEMU alerts evaded the reference engine;

    a. the first was humanly verified to be an attack, though possibly detectable by means other than line-wire IPS,
    b. the second prompted a new reference engine line-wire signature to be written, and
    c. the third is ambiguous.

3/2221 is well under a 1% improvement in performance.

However, from a scientific perspective, detecting such large volumes of attacks through non-signature based techniques without false positive is a very impressive accomplishment.

References

[1] M Polychronakis, K G. Anagnostakis, E P. Markatos.
Network-Level Polymorphic Shellcode Detection using Emulation.