Initial Report on IDS Suitability for the Project (Comparison)

Hendra Gunadi (Hendra.Gunadi@murdoch.edu.au)

March 9, 2017

1 Project Description

The goal of the project is to extend an open source Intrusion Detection System (IDS) to detect network-based covert channels. At the moment there are a lot of academic literatures for proposed techniques to do covert channels and how to detect them, but there is no integrated approach to detect covert channel yet. This is due to the vast possibilities of covert channel which means that it is not feasible to address each covert channel separately (huge overhead). The project is then proposed to bridge this gap between academic and public space by exploring the idea of a flexible and extensible framework inside an IDS and provide a proof of concept implementation to show the framework's usability. In addition, we will also explore the idea of generalizing covert channel techniques based on patterns which further boosts the flexibility.

2 Intrusion Detection System (IDS)

IDS is a security mechanism in which the priority is to be able to detect intrusions and deal with them accordingly. Generally, IDS is categorized as Network IDS (NIDS) or Host IDS (HIDS) which differs from each other in terms of the focus of the system. HIDS is primarily dealing with the intrusion against host system, e.g. file integrity, while NIDS is primarily dealing with intrusion in the network traffic. To some degree, HIDS may be more powerful than NIDS as HIDS reporting is more comprehensive than NIDS (it also detect anomaly in the resulting activity). Even though HIDS may monitor network traffic like NIDS, but there are several reason why we focus our attention to NIDS instead of HIDS:

- HIDS may be more easily compromised as host security can depends on end user using/maintaining the host vs NIDS runs on machine managed by professional staff.
- There are now more and more low-power/low-performance devices like smart phones and all sorts of embedded devices (TVs, CCTV cameras,

DVRs etc.) which may not be able to run a HIDS, so a NIDS has to be used to protect them.

- There are quite a number of operating systems, which means for the implementation we would either have to focus on a particular subset OS or spend extra time on porting code
- Covert channel may happen in a communication to and from the host, but not involving the host itself, e.g. the case where the covert channel sender / recipient is the immediate node from the host.

In the following we compare some of the NIDS tools that are widely used. For a more comprehensive comparison between the tools, the reader is referred to [16] (and to [6] for a more detailed performance comparison).

2.1 Snort[7]

Snort is one of the most popular open source NIDS which has been there since 1998. It works by sniffing the packet from the network, decode the packet, and then take action based on the rules specified. Snort is extensible in a sense that people can contribute to the rules easily. However, adding new mechanism to analyze network traffic is not so straightforward. If we want to add covert channel detection, then we have to incorporate the detection mechanism into the core of Snort and recompile the whole package (patches). Snort is still actively developed, with Snort 3.0 underway [8]. It's mailing lists also have recent activities [11, 9, 10].

2.2 Suricata[12]

Suricata is the direct competitor of Snort. It is a relatively new NIDS compared to Snort, works in a similar way to Snort (focusing on the rule matching), with the difference that Suricata is multi-threaded, as opposed to Snort that is currently single-threaded (Snort developers are planning for the multi-threaded version [8]). There is a more detailed comparison between Snort and Suricata in [15]. Since it is similar to Snort, it is also not that easy to extend the NIDS with covert channel detection mechanism. Suricata is still actively developed[13], and the mailing list also has recent activities [14].

2.3 Bro[3]

Bro is another open source NIDS that takes on different angle than Snort and Suricata. As opposed to matching packets against rules, it passively observe what is happening in the network and reports whatever it sees. Although Bro also works as an IDS, it does so by adding the detection / analyze plugin based on the network traffic report. With this infrastructure, the task of extending Bro is then a matter of creating another analyze plugin which will be based on the covert channel detection technique we implemented and the events generated

by Bro. Bro is still actively developed [2], it also has its own conference[4], and the mailing list also has recent activities [1]. Initially, Bro is developed as an academic tools, so prior to version 2.0 the usability is quite low. However, since version 2.0, Bro switches gear and hired dedicated software engineers to bridge the gap between academic use and operational use [5].

3 Comparison and Conclusion

Below I propose a summary of the features comparison between Snort, Suricata, and Bro.

- Goal compatibility describes whether the goal (or mechanism) of the IDS is in line with our project's goal. Snort and Suricata do not really align with our project's goal as they already have a fixed set of detection mechanism (rules matching) while we want to add a mechanism to detect covert channel (which may not necessarily be a rules matching). In contrast, Bro mainly deals with reporting events and the detection mechanism is built on top of these comprehensive reporting, which we may also tweak to include a mechanism to detect covert channel.

- Type of IDS describes the detection mechanism of the IDS. Snort and Suricata matches traffic against signature / rules, and then report the event should there be a violation of policy. Bro, on the other hand, works by analyzing the events reported. Depending on the detection plugin, it can be anomaly based or rule based (Bro has a signature framework), or even based upon statistical analysis of the events. This make Bro a flexible tool.

- Extensibility describes what involves in extending the IDS with a new covert channel detection mechanism. Since Snort and Suricata already have a fixed detection mechanism, to incorporate a new detection mechanism it means we have to add a new mechanism to the core IDS. For Bro, we just need to add a new plugin which makes use of the events reported to determine whether there is a covert channel in the communication. Note that at this moment we are relying on the reporting mechanism in Bro, so if Bro doesn't have the events that we are interested, then it also involves adding new events to report in Bro.

- Scalability describes whether the IDS is easily deployed to handle big network traffic. Snort is going for multi-threaded version in Snort 3.0 , but the released version 2.9 is still a single-threaded program. Suricata has been a multi-threaded program so it is easier to scale and making use of multi core CPU compared to Snort (although it doesn't necessarily mean Suricata is more superior). Bro is in itself a single threaded program. Nevertheless, Bro can be designed as a cluster of Bros that can communicate with other instance of Bro. This feature of Bro makes it scalable even

when the implementation of covert channel detection is computationally heavy.

- Active feature describes whether there is a recent development and / or mailing list activity.

Feature	Snort	Suricata	Bro
Goal compatibility	No	No	Yes
Type of IDS	Signature	Signature	Flexible
Extensibility	New mechanism	New mechanism	New plugin
Scalability	Single-Threading	Multi-Threading	Hybrid
Active	Yes		

Based on the comparison, I propose that we are working with Bro as the open source target IDS that we extend. The main reason is because Bro is closer to the goal of the project and Bro is more flexible than Snort and Suricata.

References

[1] Bro mailing list archive.

[2] Bro network monitoring project. `https://github.com/bro`, Accessed: 8 March 2017.

[3] The bro network security monitor. `https://www.bro.org/`, Accessed: 8 March 2017.

[4] Brocon'17.

[5] Liam randall- shmoocon 2013: Bro ids and the bro network programming language.

[6] Open source ids high performance shootout. `https://www.sans.org/reading-room/whitepapers/intrusion/open-source-ids-high-performance-shootout-35772`, Accessed: 8 March 2017.

[7] Snort - network intrusion detection & prevention system. `https://www.snort.org/`, Accessed: 8 March 2017.

[8] Snort 3. `https://www.snort.org/snort3`, Accessed: 8 March 2017.

[9] Snort developers mailing list archive. `http://marc.info/?l=snort-devel`, Accessed: 8 March 2017.

[10] Snort rules mailing list archive. `http://marc.info/?l=snort-sig`, Accessed: 8 March 2017.

[11] Snort user mailing list archive. http://marc.info/?l=snort-users, Accessed: 8 March 2017.

[12] Suricata | open source ids / ips / nsm engine. https://suricata-ids.org/, Accessed: 8 March 2017.

[13] Suricata activity, note = https://redmine.openinfosecfoundation.org/projects/suricata/activity, accessed: 8 march 2017.

[14] Suricata mailing list archive, note = https://lists.openinfosecfoundation.org/pipermail/oisf-devel/, accessed: 8 march 2017.

[15] Suricata-vs-snort. https://www.aldeid.com/wiki/Suricata-vs-snort, Accessed: 8 March 2017.

[16] Suchita Patil, Pallavi S Kulkarni, Pradnya B Rane, and BB Meshram. Snort, bro, netstat, emerald and sax2: A comparison. *International Journal of Advanced Research in Computer Science*, 3(2), 2012.