Comparison of IDS Suitability for Covert Channels Detection

Hendra Gunadi (Hendra.Gunadi@murdoch.edu.au),
Sebastian Zander (s.zander@murdoch.edu.au)

August 18, 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 for proposed covert channels techniques and how to detect them, but there is no integrated approach to detect coverts 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 will 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 detection techniques based on patterns [1] which further boosts the flexibility.

2 Intrusion Detection System (IDS)

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

- A HIDS may be more easily compromised as host security depends on end users using/maintaining the host whereas a 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 of OS or spend extra time on porting code between OS when using a HIDS.
- Covert channels may happen in a communication to and from a host, but not involving the host itself, e.g. the case where the covert channel sender / recipient is an immediate node from the host, and so cannot be detected by a HIDS.

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 [2] (and to [3] for a more detailed performance comparison).

2.1 Snort [4]

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, decoding the packet, and then taking actions based on the rules specified. Snort is extensible in a sense that people can contribute to the rules easily. However, adding new mechanisms 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 (ie. provide our code as patches to snort). Snort is still actively developed, with Snort 3.0 underway [5]. It's mailing lists also have recent activities [6–8].

2.2 Suricata [9]

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 a multi-threaded version [5]). There is a more detailed comparison between Snort and Suricata in [10]. Since Suricata is similar to Snort, it is also not that easy to extend it with a covert channel detection mechanism. Suricata is still actively developed[11], it also has its own conference[12], and the mailing list also has recent activities [13].

2.3 Bro [14]

Bro is another open source NIDS that takes on a different angle than Snort and Suricata. As opposed to matching packets against rules, Bro 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 covert channel detection techniques we will implement and the events generated by Bro. Bro is still actively developed [15], it also has its own conference [16], and the mailing list also has recent activities [17]. Initially, Bro was developed as an academic toolset, so prior to version 2.0 the usability is quite low. However, since version 2.0, Bro switched gear and hired dedicated software engineers to bridge the gap between academic use and operational use [18].

3 Comparison and Conclusion

Below we compare Snort, Suricata, and Bro based on a number of important features. Table 1 summarizes the comparison.

- *Type of IDS* describes the detection mechanism of the IDS. Snort and Suricata match traffic against signatures / rules, and then report an event in case of a violation of policy. Bro, on the other hand, works by analyzing the events reported. Depending on the detection plugin, Bro can be anomaly based or rule based (it has a signature framework), or even based upon statistical analysis of the events. This make Bro a flexible tool.
- *Extensibility* describes how the IDS can extended with new covert channel detection mechanisms. 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. To be more precise, Snort also has a mechanism to add new detection plugins but it is limited to add new keywords for the rule matching and which parts of the packet to inspect. 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.

Table 1: Comparison of Snort, Suricata and Bro

Feature	Snort	Suricata	Bro
Type of IDS	Signature	Signature	Flexible
Extensibility	New mechanism	New mechanism	New plugin
Capacity per Instance	> 1 Gbps[19]	> 1 Gbps[20]	250 Mbps[14]
Scalability	Single-Threading	Multi-Threading	Hybrid
User Base	++	+	+
Active Development	Yes		
License	GPL v.2	GPL v.2	BSD

- *Capacity per instance* describes the suggested bandwidth that can be handled by each instance of the NIDS. Even though Bro has lower bandwidth than Snort [19] and Suricata [20], in a common 4-core CPU Bro can reach a reasonable bandwidth of 1 Gbps [14].
- *Scalability* describes whether and how the IDS scales with large network traffic volumes. Snort is going for a 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 CPUs compared to Snort (although it doesn't necessarily mean Suricata is superior). Bro is in itself a single threaded program. Nevertheless, Bro can be deployed as a cluster of Bros that can communicate with each other. This feature of Bro makes it very scalable even for tasks that are computationally heavy, such as covert channels detection.
- *User base* indicates that Snort is the most popular and widely used NIDS at the moment. That said, both Suricata and Bro also have their own user base in their respective conferences.
- *Active development* describes whether there is a recent development and / or mailing list activity. All three IDS are under active development.
- *License* described the license under which the NDIS is released. All of the NIDS use free license. Bro's license is the most liberal (BSD license).

Based on the comparison, 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 channels (which cannot solely be handled by rules matching). In contrast, Bro mainly deals with reporting events, and has a very flexible architecture that allows different types of plugins which can report events, which we can tweak to include mechanisms to detect covert channels. Bro also has reasonable bandwidth, and scales well due its cluster architecture. On top of that Bro will also be multi-threaded in the next version, which should further improve its performance and scalability. While Bro is not the most popular IDS, it has a decent number of users and there is significant development activity. Bro is open source and release under a the liberal BSD license.

Based on the comparison we propose to use Bro as the open source IDS that we will extend with covert channels detection.

Acknowledgements

This work was supported by a grant from the Comcast Innovation Fund.

References

[1] Steffen Wendzel, Sebastian Zander, Bernhard Fechner, and Christian Herdin. Pattern-based survey and categorization of network covert channel techniques. *ACM Computing Surveys*, 47(3):50:1–50:26, April 2015.

[2] 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.

[3] 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.

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

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

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

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

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

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

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

[11] Suricata activity. https://redmine.openinfosecfoundation.org/projects/suricata/activity, Accessed: 8 March 2017.

[12] Suricon | 2017 suricon in prague presented by oisf. https://suricon.net/, Accessed: 10 March 2017.

[13] Suricata mailing list archive. https://lists.openinfosecfoundation.org/pipermail/oisf-devel/, Accessed: 8 March 2017.

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

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

[16] Brocon'17. https://www.bro.org/community/brocon2017.html, Accessed: 8 March 2017.

[17] Bro mailing list archive. http://mailman.icsi.berkeley.edu/pipermail/bro/, Accessed: 8 March 2017.

[18] Liam randall- shmoocon 2013: Bro ids and the bro network programming language. https://www.youtube.com/watch?v=7DCPuHdCbpw, Accessed: 8 March 2017.

[19] Pf ring 10 gbps snort ids. https://metaflowsblog.wordpress.com/2013/09/26/and-now-for-something-completely-technical-pf-ring-10-gbps-snort-ids/, Accessed: 10 March 2017.

[20] Suricata, to 10gbps and beyond. httpsa://home.regit.org/2012/07/suricata-to-10gbps-and-beyond/, accessed: 10 March 2017.