

OISF

The Open Information Security Foundation

Planning and Technical Meeting Summary

Washington DC, July 16th, 2009

The technical meeting went very well. Our sincere thanks to everyone that attended and contributed to the discussion. We filled our second larger room to capacity and unfortunately had to turn away attendees at the last minute. The discussion was very productive. We have gained a number of new Consortium members and are talking to several more to solidify arrangements.

The key element we have taken away from the meeting is to limit our scope to the primary advancements and core features initially, and then build upon this foundation as it stabilizes. OISF will take a more serial approach to development as opposed to our previous parallel initiatives using coders independently. We are incorporating this excellent advice into our development time lines and tasking immediately.

Additional important elements of the technical discussion were the requests for extensive APIs to allow external interaction. Allowing for these external application interactions will be a significant goal for all modules, input and output..

What follows is a summary of other major discussions at the meeting. Finalized Phase One and Phase Two objectives documents will be available soon.

Administrative

1. OISF will stagger the board members election years/dates in an effort to maintain consistency in governance. Initial nominations for Board of Directors positions will happen soon.
2. Working Groups will be formed to address some of the more complex issues. Volunteers will be requested, put into mailing lists and a group leader/moderator will be appointed from the working group. Each group will receive a due date for their topic deliverable.

Consortium

1. **Endace** has committed to being a Consortium Member and supporter of OISF. Endace has already shipped several traffic generation boxes and 10gig capture cards to our QA lab in Kansas City. Endace will also provide two programmers in a near full time capacity to the foundation. These programmers have significant capture and hardware expertise as well as high speed processing experience.
2. **Breach Security** has committed to being a Consortium Member and supporter of OISF. Breach will provide half time access to Brian Rectanus who has extensive knowledge of HTTP security parsing and detection via Mod_Security, and many other valuable software development skills. .

3. **Nitro Security** has committed to being a Consortium Member and supporter of OISF. We are in talks to determine their contribution from several options available.
4. **Everis Security** has committed to provide coding support and access to their extensive test bed facilities.
5. OISF is talking to several other companies about how we can work together. More news on at least 6 new partnerships in the coming weeks!
6. OISF has offers from several agencies and commercial companies to recreate and assist with QA and testing in their labs to confirm or reinforce our efforts.

Technical

1. OISF will **develop the engine to the x86 architecture** initially, but will consider others as demand is presented from the community. Assistance and hardware from specific vendors will be required in the more unusual architectures.
2. Rules will have an additional directive to **support external code**. The languages to support will be determined in Phase Two.
3. **IPv6 decoding and detection** is a requirement for Phase One.
4. The engine will be developed and QA'd with **10gigabit speeds as our production release goal**. QA and performance testing methodologies are in development.
5. The engine will be **QA'd on virtual platforms** (VMWare, QEMU, etc) to ensure virtual sensors are effective. Discussion of streaming captured packets to guest OSs via memory mapping will be explored.
6. **IP Reputation** is a Phase One feature. A working group will be established to determine categories of reputation to be stored. This group will also explore domain name reputation and url reputation inclusion in the same stream and associated tools.
7. There is a lot of interest in a **statistical DDoS detection module** and less for a portscan module. The Foundation will seek input from domain experts regarding the best way to do both of these. This may be a Phase Two feature.
8. **OpenCL and Cuda** are being investigated for acceleration using stock graphic cards. There was a lot of positive interest in this to augment standard hardware. This will be implemented in Phase One if the working group determines this is a feasible deliverable.
9. OISF will consider **Indexing for Capture Files**. Specifically, the engine should index stream locations and file designation to allow later retrieval by systems conducting disk capture. Endace has significant experience with this functionality and has offered to assist in coding via their contributed coders. This will be a Phase Two objective.
10. **Event Output** will be handled in a binary stream where possible. Post processing will be done by a separate tool. OISF will maintain backwards compatibility where possible with Snort output and schemas, but will develop other methods as there will be significantly more and different output from this tool versus Snort. MITRE representatives noted that they have a new event description language due to be released within a few weeks. OISF will consider this as a standard to implement.
11. **Engine Configuration can** be stored in the database as well as fed to the engine through a text file. Database storage will allow easy centralized management of large numbers of sensors. An API

is a requirement for Phase One. A configuration manager is a potential Phase One release, but no later than a Phase Two feature.

12. OISF will build a **Web Configuration Manager** running on a light http engine. This will allow option and drop down configuration of the engine configuration to be inserted into the database, or generation as a text file. This is a lesser priority item but likely to be implemented in Phase Two. An API to access this configuration database is a requirement.
13. OISF will consider building a **Rules and Event manager** to be included with the engine and integrated into the configuration manager. This would be a tool with minimal functionality for the smaller scale user and environment. BASE is being considered as a tool to be rewritten into the engine release. Ruleset and configuration version control is a requirement. This will have an accompanying API to allow other tools easy access. This will be a Phase Two objective.
14. **Live Rule Reloading** will be investigated as an option in the engine. Many environments require this functionality. This may be a Phase Two feature.
15. The engine will create a **Global State Database** that rules and plugins will access. This will allow global flowbit style functionality and the description of events between streams. This database will store data in variables and in states. This is a Phase One feature.
16. **Global State** will persist between rule reloads and engine starts and stops. If this option is enabled, information will be written to disk on engine shutdown.
17. The Engine will have the ability to **rollback to a previous configuration** if loading fails. This will be available as an option whether using live rules reload or not. This is a Phase One priority.
18. **Community Ruleset Language** preference is to the development of a new language that will support the import of Snort and other syntax rules. A Phase One goal will be to support the use of Snort syntax rules but using a superset of directives to describe things like scoring and IP Reputation. Phase Two will include creating a new language and implementing it as a primary language for the engine with Snort importing. A working group will explore this objective.
19. **Rule Obfuscation** is a Phase Two feature. While not ideal OISF realizes some form of rule obfuscation is necessary to hide the content of rules regarding undisclosed vulnerabilities. A working group will be established to explore the necessity of this and a method (compilation, encryption, etc).
20. **Scoring Thresholds** will be a Phase Two feature. This will allow a rule writer to add points to an attacker or stream similar to SpamAssassin style spam scoring. Once a stream or attacker crosses a threshold based on a group rules the stream can be flagged or blocked. The local administrator defines the thresholds and actions.
21. **Passive Fingerprinting of Applications and Operating Systems** will be implemented as a Phase Two feature. Rules will be able to access information about IPs involved in a stream via the Global State Database.
22. **Protocol Identification based on traffic** (as opposed to ports) will be implemented in Phase One. This will allow rules to specify ports to monitor and/or protocols detected.
23. **Updates to sensors should be parallelized** via a flag in the configuration database to update. This feature enables users to specify sensors as “Beta” sensors. The Beta sensors would implement a new configuration and report a success or failure in loading. Other sensors would then load configurations or rulesets after at least one or all Beta sensors reported success. This is a Phase One feature.

24. The **configuration file will use a standard syntax throughout**. This will allow easier parsing and writing. This is a Phase One feature.
25. The engine will optionally write **statistics to a centralized database** or local file on a defined interval. These statistics will include things such as throughput, protocol distribution, packets dropped, stream statistics, system load, etc. This module may also be configured to include information about the engine's internal processes to identify modules or rules that are not functioning at expected efficiency levels. This is a Phase Two feature.
26. The engine will be built as an **Embeddable Library** to enable use in other applications. This will allow subsets of functionality to be used easily in other tools. This is a Phase One feature.
27. **Multi-Threading** is a high priority Phase One feature. This is feature is nearly complete.
28. **Hardware Acceleration** support will be implemented for every platform for which the foundation receives support. Endace and Bivio have pledged support to make this possible on their platforms.

Working Groups

1. **DDoS and Portscan Statistical Analysis Methods**. This group will explore state of the art detection of inbound and outbound DDoS Detection, as well as better ways to detect portscanning (classic, vertical, horizontal and distributed). DDoS analysis will be focused on all forms including (but not limited to) HTTP, ICMP, UDP, and other protocols.
2. **OpenCL/Cuda Acceleration**. This group will investigate using either or both of these technologies for off the shelf graphics cards as acceleration devices. Output from this working group will be a decision regarding the feasibility of implementation, which technologies to use, and any other relevant topic research discoveries.
3. **Rules Language**. This group will determine which Snort syntax directives are used frequently enough to be implemented, what directives must be added to support new functionality, and determine whether a new rules language is warranted.
4. **IP Reputation**. This group will define the categories to be tracked in the reputation module(i.e. spammer, CnC, scanner, open proxy, etc). This group will explore the feasibility of including in stream reputation information about domain names and hostile URLs. A scoring methodology will also be defined. Output will be the categories and scoring methodology and a feasibility determination for domain and URL tracking.
5. **Rule Obfuscation**. This group will explore obfuscating rules about undisclosed vulnerabilities. While this functionality is not ideal in an open source security community, it may be necessary to enable the use of data from sources that do not allow disclosure of rule content for certain periods of time. This group will explore the necessity of this feature for the engine. If it is determined that this functionality is required, the group will also identify methods for implementing obfuscation (encryption, compilation, etc).
6. **Fast Flux Detection**. This group will explore the feasibility and criticality of a domain fast flux detection module. If required relevant research will be provided.