Merlin SRP - Project Update

Michael Himbeault

November 13, 2009

1 Where We Came From

1.1 Literature Review

- Several papers on network payload signature based analysis. This is problematic for several reasons, the least of which is that it has difficulty handling encrypted payloads. It often requires the reverse engineering of network and communication protocols used by the specific malware in question necessitating that a copy of the program be on hand for analysis. This isn't particularly reasonable for Merlin's situation as the neither the expertise nor the time are available to devote to this method.

  Additional problems include lack of scalability (For large networks with multiple different 'strains' of infection, this process must isolate each strain individually), portability (It doesn't port well from location to location) and adaptability.

  For these reasons, this approach has been largely discarded.

- As was mentioned last time, the concept of "social interaction" of network hosts is an approach that is agnostic to payload schemes (But may care about payload size). This approach is not unknown to literature in this area and such graphs are known as Traffic Dispersion Graphs (TDGs). The originating paper appears to be [3] and discusses possible analysis of the resulting graphs. These graphs were captured by monitoring a network backbone and considering the resulting data.

  A paper exists([2]) that looks at using TDGs for the purpose that was suggested in October; detecting malware based on the social characteristics of its network activity. This paper indicates that if the malware in question implements some simple peer-to-peer algorithms it is possible for it to reduces its footprint in a TDG-based analysis to such a level as to almost disappear into the background noise of normal activity. The scenario that this paper considers is significantly different form the scenario in the Merlin project for several reasons and so these results, while instructive, are not considered completely valid to this project.

1.2 Thoughts on Approaches

Malware, in particular botnets, have several unchanging properties; the most useful of which is that each bot must be able to receive orders from a controller. It is this property that probably is least likely to change with future strains. How can it receive these orders?

Hardcoded: The address of the controller can be hardcoded into the malware. This is highly unlikely as it would make the controller very vulnerable to discovery should the source code be reverse engineered to obtain this address. This method is considered to not be in use for this reason.

DNS Lookups: The only other option for direct communication requires a DNS lookup. E.g: Conficker.

Instant Messaging: e.g: MSN, Yahoo, AIM

IRC: A common choice due to its ability to run on almost any port (including the often forwarded port 80), and the availability of hosted servers.

HTTP Hosted Services: Twitter.

Indirect: It could be possible that not all bots know where the controller is, and so orders filter through a peer-to-peer bot network.

Each of these have their own distinct signatures that should be detectable in the network traffic of a sufficiently large botnet.

2 Where We Are Going

Literature review: This is never complete, but a preliminary survey is complete insofar as it has produced information on the most common approaches.

Get data: Jared is supplying the data at a rate well beyond what can be used at this time but will be useful for 'back analysis' to obtain some sort of frame of reference for when analysis begins to scale up.

Look at data*: The best pattern matching tool available is the human brain. It is desirable to use a visual framework to be able to visualize the data that is being produced. The problem is that out of the several tools evaluated none can accomplish the needs that are required (Massive data sets, run-time manipulation of visualization and underlying data): So a custom visualization API with a focus on exploration of massive data sets is in development.

Hypotheses: From the data exploration, determine some possible rules and heuristics for identifying unusual behaviour.

Magic: ...

3 Where We Are

The approach that is expected to yield the best results is a hybrid approach:

Consider the DNS lookup patterns based on the results in [1]. Jared is working on collecting this information (As it is not contained in the flow data).

Based on the results of the DNS query analysis, consider some social properties of some subset of hosts specified by the DNS analysis.

It is expected that subnet-grained data is going to be insufficient for TDG analysis.

Possible: Consider the payload along some questionable links as defined in the previous step. This may or may not be of use (Could attempt to detect encrypted data, or something...).

References

[1] David Dagon, *Botnet detection and response: The network is the infection.*

[2] Mark Jelasity and Vilmos Bilicki, *Towards automated detection of peer-to-peer botnets: On the limits of local approaches.*

[3] Marios Iliofotou, *Network Monitoring using Traffic Dispersion Graphs TDGs.*