BOTWATCHER
Transparent and Generic Botnet Tracking Complementary Material

Thomas Barabosch[1], Adrian Dombeck[1], Khaled Yakdan[1,2], Elmar Gerhards-Padilla[1]

[1] Fraunhofer FKIE, Germany
{firstname.lastname}@fkie.fraunhofer.de
[2] University of Bonn, Germany
yakdan@cs.uni-bonn.de

This *zip* file contains complementary material of the paper BOTWATCHER*: Transparent and Generic Botnet Tracking*. On the one hand side, we included the data of the evaluation as *csv* files. On the other hand side, we included further inference rules of BOTWATCHER (cf. next section).

1 Inference Rules

At the core of BOTWATCHER's correlation logic are inference rules represented by the form:

$$\frac{P_1\ P_2\ \ldots\ P_n}{C}$$

The top of the inference rule bar contains the premises $P_1, P_2, \ldots, P_n$. If all premises are satisfied, then we can conclude the statement below the bar C. Inference rules provide a formal and compact notation for single step inference and implicitly specify an inference algorithm by recursively applying rules on premises until a fixed point is reached. BOTWATCHER uses an extensible set of rules that cover a wide spectrum of malware-related actions. We show current inference rules in Figure 1. See the paper for a detail description of the symbols. Note that we will enhance this table with new inference rules from time to time (current state: September 8, 2015).

$$(1)\ \frac{\mathcal{M}\left[a_s, a_e, p_{id}, ts_1\right] \quad \mathcal{T}\left[t_{id}, p_{id}, a_t, \tau_{\mathbf{create}}, ts_2\right] \quad ts_2 \geq ts_1 \quad a_t \in [a_s, a_e]}{\textsc{CodeInjection}\left[p_{id}, ts_2\right]}$$

$$(2)\ \frac{\mathcal{N}\left[s, p_{id}, \tau_{\mathbf{start}}, -, ts_1\right] \quad \mathcal{N}\left[s, p_{id}, \tau_{\mathbf{stop}}, -, ts_2\right] \quad ts_2 \geq ts_1}{\textsc{Download}\left[p_{id}, ts_2\right]}$$

$$(3)\ \frac{\textsc{Download}[pp_{id}, ts_1] \quad \mathcal{P}\left[p_{id}, pp_{id}, \tau_{\mathbf{create}}, ts_2\right] \quad ts_2 \geq ts_1}{\textsc{MalwareDownload}\left[p_{id}, ts_2\right]}$$

$$(4)\ \frac{\textsc{Download}[p_{id}, ts_1] \quad \mathcal{M}\left[-, -, p_{id}, \texttt{rwx}, ts_2\right] \quad ts_2 \geq ts_1}{\textsc{MalwareDownload}\left[p_{id}, ts_2\right]}$$

$$(5)\ \frac{\textsc{Download}[p_1, ts_1] \quad \textsc{CodeInjection}\left[p_2, p_1, ts_2\right] \quad ts_2 \geq ts_1}{\textsc{MalwareDownload}\left[p_2, ts_2\right]}$$

$$(6)\ \frac{\mathcal{N}\left[s, p_{id}, \tau_{\mathbf{start}}, -, ts_1\right] \quad s.\texttt{dest} \in PM}{\textsc{Spamming}\left[p_{id}, ts_2\right]} \qquad (7)\ \frac{\mathcal{C}\left[ts, k, v\right] \quad k \in K}{\textsc{Persistence}\left[p_{id}, ts_2\right]}$$

$$(8)\ \frac{\mathcal{N}\left[s, p_{id}, \tau_{\mathbf{start}}, -, ts_1\right] \quad s.\texttt{status} = \texttt{listening}}{\textsc{CCInfrastructure}\left[ts\right]}$$

$$(9)\ \frac{\mathcal{N}\left[s, p_{id}, \tau_{\mathbf{start}}, -, ts_1\right] \quad \Delta\left[s\right] \geq T_{min}}{\textsc{CCInfrastructure}\left[ts\right]}$$

$$(10)\ \frac{\mathcal{N}\left[s_i, p_{id}, \tau_{\mathbf{start}}, -, ts_i\right]^{i \in 1..k} \quad \Omega\left[s\right] \leq \omega_{max}}{\textsc{CCInfrastructure}\left[ts\right]}$$

$$(11)\ \frac{\mathcal{K}\left[-, ts\right]}{\textsc{Rootkit}\left[ts\right]}$$

Fig. 1: Exemplary inference rules. Premises are execution events and as conclusion malicious behavior is inferred.