THERMINATOR 2: A THERMODYNAMICS-BASED METHOD FOR REAL-TIME PATTERNLESS INTRUSION DETECTION

Stephen D. Donald, Robert V. McMillen, David K. Ford, and John C. McEachen
Re-configurable Intrusion Detection Laboratory Research (RIDLRR)
Department of Electrical and Computer Engineering
Naval Postgraduate School
437 Dyer Road, Code EC/Mj
Monterey, California 93943, USA
mceachen@ups.navy.mil

ABSTRACT
A novel system for conducting non-signature based, or patternless, intrusion detection of computer networks is presented. The initial prototype has been installed at U.S. Pacific Command and Army Signal Command. This system uses principles of thermodynamics to model network conversation characteristics. Observing the properties of entropy, energy and temperature within the system develops a notion of baseline operating conditions. Perturbations in these properties are considered potential intrusions for further investigation. System functions are decomposed into a network sensing device, a real-time processing component and a forensics component. State definitions for a variety of conditions are discussed. Finally, examples of valid intrusions and other network perturbations in real traffic collected in network operation center backbones are presented. Preliminary results indicate this system has significant potential for revealing anomalies in large network systems.

INTRODUCTION
By definition, an intrusion is somebody attempting to break into or misuse your system. The word "misuse" is broad, and can reflect something as severe as stealing confidential data to something benign such as misusing network resources. An Intrusion Detection System (IDS) is a system for detecting such intrusions [1].

An IDS' primary means of identifying an intrusion is by the use of signatures. Signatures are predefined patterns that may indicate a specific attack or anomaly. A signature based IDS monitors network traffic searching for bit patterns that match its database of signatures.

The use of signature patterns to identify intrusions is a quick and efficient process at the expense of flexibility and adaptability [2]. IDS developers constantly produce signature updates because, if a signature is not in the database, the IDS will not detect the intrusion. Also, due to the availability of malicious software on the Internet, anyone with sufficient knowledge in computer programming can make quick modifications and produce undetectable malicious code.

This paper presents a method that will potentially allow a network analyst to use other means (non-signature based) to detect and identify intrusions. The process described in this document does not depend on signatures. Rather, it uses a mathematical model [3] to graphically represent the network's general characteristics using thermodynamic principles such as energy, entropy, and temperature.

Burgess [4] has presented similar research that models computer networks using the concepts of statistical physics. While his paper provides a hypothetical analysis, he does not provide any applied analysis and makes no consideration for actual implementation.

The intent of this system, dubbed Therminator, is not to replace existing IDS technology but complement it. The combination of the two will hopefully benefit from the routine update of the commercial IDS signature database, which provides speed and efficiency. Ideally then, when Therminator identifies an event not known to the IDS, a new signature can be developed by using the data acquired by the Therminator sensor. Also, the "false-alarm" rate (IDS identifies a friendly event as hostile) can be reduced by correlating the output of the two systems. This correlation will ensure the analyst can distinguish an IDS attack from an actual attack on the network.

THERMAL PROPERTIES OF NETWORK TRAFFIC
As was stated before, the goal of Therminator is to reduce standard network data into a useful, reproducible, and meaningful form, ultimately to allow accurate detection of network anomalies. To do this, the statistical mechanics process is going to be reversed, entering the process with a notion of
continuously observed microscopic state transitions and exiting with macroscopic notions of entropy, energy, temperature, work and heat. The notion of state will be more carefully defined below, but quickly summarizes into activity levels of various conversation groups within the network. The Hamiltonian is, in effect, the aggregate set of rules that govern the conversations, including protocol interactions as dictated by the Internet Engineering Task Force’s (IETF) Request for Comments (RFCs), actual and logical network architecture, and such amorphous entities as security restrictions and usage policies. It is however the state transitions that are observed. One end product is a real time graphical description of the configuration of the network.

The state model used by Thermator tries to model the typical characteristics of network traffic while providing a usable notion of state for the thermodynamic calculations. At the most basic level, Thermator’s view of network state is a collection of network conversation groups and their respective relative information levels. A state transition will be an exchange of information elements between the involved conversation groups. The analogy used during development describes the conversation groups as buckets and the information element as balls. A bucket will contain some ordinal number of balls, and the collection of buckets and their respective number of balls will be the state of the network. The bucket can be defined using any combination of conversation characteristics including who is talking (individual hosts or networks), the language they are speaking (TCP, UDP, or ICMP), or the job they are performing (client or server). The implemented model can use any field within the sensed data to define conversation groups. A ball is a relativistic representation of the information a bucket contains. A state transition causes a shift in the distribution of information between the buckets. This model translates network behavior into state transitions by selecting a ball from the bucket matching the source characteristics of the packet and moving that ball into the bucket matching the destination characteristics of the packet, thereby redistributing the information and transitioning the state.

For example, bucket A defines the conversation group containing hosts X and Y, while bucket B defines the conversation group containing host Z. Each bucket will be initialized with 3 balls yielding a starting state of \( \{3, 3\} \). For this example, the starting state is arbitrary, a more detailed analysis of starting state and boundary selections will be discussed below. The following information exchanges are recorded: transiting the network: \( Y \rightarrow Z, Z \rightarrow Y, X \rightarrow Z, Y \rightarrow Z, \) and \( X \rightarrow Y \). Applying the Thermator state transition model, upon receipt of the first \( Y \rightarrow Z \) exchange a ball is selected from bucket A, which matches the source characteristics of the exchange. That ball is then placed into bucket B, which matches the destination characteristics of the exchange. The result is a transition to a new state of \( \{2, 4\} \). The next exchange, \( Z \rightarrow Y \), removes a ball from bucket B and places that ball into bucket A, resulting in a revisit to the state \( \{3, 3\} \). \( X \rightarrow Z \) removes a ball from bucket A and places it into bucket B revisiting the \( \{2, 4\} \) state. \( A \rightarrow Z \) exchange occurs again resulting in a transition to the new state \( \{1, 5\} \). Finally, the \( X \rightarrow Y \) exchange repeats the previous state \( \{1, 5\} \), because the state transition model causes the ball to be removed from bucket A and then placed back into bucket A. In summary, the exchanges \( Y \rightarrow Z, Z \rightarrow Y, X \rightarrow Z, Y \rightarrow Z, \) and \( X \rightarrow Y \) result in the states \( \{2, 4\}, \{3, 3\}, \{2, 4\}, \{1, 5\}, \) and \( \{1, 5\} \) when given an initial state of \( \{3, 3\} \).

Figures 1 and 2 show the importance of the state walk, and how the model provides more information than a simpler model. Both graphs represent a state space made up of two conversation groups, bucket A and bucket B, and they start with 5 balls each. As stated before, a network exchange can cause a state transition in three ways: by moving a ball from bucket A to bucket B, by moving a ball from bucket B to bucket A, or by either moving a ball from bucket A to bucket A or bucket B to bucket B, essentially resulting in no change for that period.

The paths shown in the figure 1 and 2 share a few notable similarities. The paths begin and end at the same state and undergo the same number and types of state changes, including five changes up, seven down, and four no changes. However, the transitions that make up those changes produce remarkably different paths, shown on the left hand side of the figures, and those paths produce remarkably different state counts, shown on the right hand side of the figures. Thus it is shown that this model provides a number of unique discriminators including number and type of state changes, starting and ending point and the general path in-between, and the state counts.

Once the states are defined, the next step in the Thermator process is to determine the individual probability of each state. Each state will have a total time in which the network occupied that state and the number of times that the state occurred. The probability, \( P \), for a given state simply equals
the time, \( k \), dedicated to that state over the sum of
the times associated with all the states during a given
measurement period.

Further, if \( m \) is the total number of unique
states in the system, and \( \pi_k \) is the number of
occurrences of state \( k \) over the entire measurement
period, then the "rate" of occurrences of a given
state during the measurement period, \( q_k \), is \( \pi_k \) over
the time in state \( k \). The significance of \( P_k \) and \( q_k \)
will be developed in the next section.

Once a probability measure is assigned to a
state the system entropy is determined by computing
the partial statistical entropy for each individual
element and then summing all of the partial
entropies to achieve the system entropy during a
discrete measurement period, \( t \), as shown in equation
(1):

\[
\text{entropy}_i = \sum_{k=1}^{m} P_k \ln P_k
\]  \( (1) \)

The state-based statistical nature of this
system, where transitions between states may take
place at any instance of time, can be represented as a
continuous time Markov chain. In this case, \( q_k \)
represents the rate of transition through state \( k \) and
the generator of the process, \( Q \), is a function of the
discrete transition probability matrix associated with the
Markov chain.

In the form of Maxwell ([6]), \( P_k \) can be
alternatively represented as:

\[
P_k = e^{-\frac{\mu_i}{T}}, i = 1, 2, \ldots, m_i
\]  \( (2) \)

and in keeping with the traditional notation in
statistical mechanics refer to \( H_i \) as the energy of the \( i^{th} \)
state and \( T \) as the system temperature. If we
carefully choose the state assignments such that the
"center of mass" of the state energies sum to zero,
we can find \( H_i \) by:

\[
H_i = T \ln \left( \prod_{j=1}^{m_i} (P_j)^{\frac{1}{m_i}} \right)
\]  \( (3) \)

In practice, the numerator rapidly
approaches zero and must be appropriately scaled by
applying an exponentiation of \( m_i \) to the final
numerator product in order to ensure computationally valid values. This scaling produces
equation (4), which more accurately reflects the
internal formula used in the Therminator process:

\[
H_i = T \ln \left( \prod_{j=1}^{m_i} (P_j)^{\frac{1}{m_i}} \right)
\]  \( (4) \)

Energy depends on the summation of the
individual states' energies, or \( H_s \), weighted by the
probability associated with that state. The total
energy of the system can thus be described with
equation (5):

\[
\text{energy}_i = \sum_{k=1}^{m} (P_k H_k)
\]  \( (5) \)

Computing system temperature is a great
deal more complex. The basis for temperature is a
transformation function comparing the state
progression of the current measurement period with
the state progression of the previous measurement
period and then relating the previous temperature to
the new temperature via this transformation. This
sets up a recursive relationship for temperature that
behaves as detailed in the equation below, where \( \phi \) is
the transformation function.

\[
T_0 = 1, \ T_t = \phi T_{(t-1)}
\]  \( (6) \)

The definition of \( \phi \) used in the Therminator
process is dependent on the state progressions \( m_i \) and
\( m_{(t-1)} \). The state progressions must be sufficiently
comparable to ensure the underlying assumptions
hold true in order to define \( \phi \) in the manner
discussed below. If the state progressions are
sufficiently similar, then the following relationship
can be shown to hold:

\[
\prod_{k=1}^{m_i} \pi_k = \left( \prod_{k=1}^{m_{(t-1)}} \pi_k \right)^{\frac{1}{\phi}}
\]  \( (7) \)

The interested reader may find a detailed derivation
of equation (7) in [3].

Simplification of the logarithm of a product
produces a summation of individual logarithms,
producing values better suited for computer
calculations.

\[
\log \left( \prod_{k=1}^{m_i} \pi_k \right) = \sum_{k=1}^{m_i} \log \pi_k = M
\]  \( (8) \)

Making the substitutions into the above
equations, the temperature at time \( t \) can be calculated
using the following equations:
\[ T_0 = 1, \ M_0 = 1, \ T_i = T_{(i-1)} \left( \frac{M_i}{M_{(i-1)}} \right) \] (9)

Figure 3 illustrates the components of the graphical display provided by the Therminator program. In the upper left hand corner is the window containing a graph of the computed entropy, energy, and temperature. The upper half of the main window, sometimes referred to as “Thermal Towers”, shows the average number of balls in a bucket for a given period of time, while the lower half of the main window, referred to as the “Thermal Canyon,” shows the sorted state counts produced during a measurement period.

The curves produced when ranking the states by their counts is analogous to the Boltzmann curves of probability versus energy state of molecules in a gas [5]. Using the Boltzmann analogy, the area under the Therminator state curve is a proportional representation of the temperature, while the slope of the state curve becomes a proportional representation of the entropy, and the change between time slices or slides becomes a proportional representation of macroscopic energy fluctuations. Changes in thermal canyon represent heat in or out of the system. Changes in network configuration reflect work done on the network by the outside world or work done on the outside network by the defended network.

By examining the manifold, or canyon, developed by collecting various curves over time, anomalies can be easily spotted as perturbations in the normal flow of the canyon. The cause of such dramatic changes in practice ranges from a single transition to many thousands of packets. The next section details some of the analysis results seen during development testing.

IMPLEMENTATION AND OBSERVATIONS

The present manifestation of Therminator was designed for IP networks. Packet assignment may be determined by any of the following conditions: matched reply, flow direction, a previously unseen IP address, IP and/or TCP header length, protocol type, packet size and broadcast indicator.

The definition of a bucket space depends entirely on the network being monitored and often requires some trial-and-error testing to develop a suitable set of conditions. Typically, three conditions (eight buckets) would be used with 20 balls.

This system has been tested at the egress point of three operational networks. Typical traffic loads ran between 3000 – 5000 packets per second. Some observed phenomena are described below.

A. ICMP Anomaly Example

Figure 4 depicts a flood of ICMP packets originating inside the monitored network after normal working hours. This flood increased the state count; thereby producing a large spike on the GUI. This flood consisted of 6,032 ICMP echo requests/replies within a ten second time frame. ICMP echo requests and replies are not necessarily anomalous, however the owner of the system was logged off and at home requiring notification of the local CERT for further investigation.

An important point to note, Therminator was the only system that captured this event. The administrators of the monitored network heavily depends on two commercial intrusion detection systems, and neither system caught the event. The only proof that helped verify the event actually happened was a router’s logs.

CONCLUSION

We have presented a system for characterizing the dynamics of conversations occurring across a boundary of an organization’s network. This system appears to have significant potential for monitoring the health of networks as well as intrusion detection. The system is modeled after statistical mechanics methods for developing notional trends from large volumes of data.

This system has been observed operating in real network environments and has produced some intriguing results. Specifically, the system has been shown to detect phenomena that are not detected by any other network sensor. While not always an indicator of malicious activity, this has proven to be a significant resource for network administrators as an aid to understanding how their network is being used.

REFERENCES

Figure 1. (Left) An example of a state walk starting with 5 balls in each bucket. (Center) A table of the transitions for each bucket. (Right) The distribution of balls in buckets for this state walk.

Figure 2. (Left) An example of a second state walk also starting with 5 balls in each bucket, however, a completely different trajectory is taken. (Center) The table of transitions for each bucket. (Right) The distribution of balls in buckets for this state walk. Note the significant difference from the right hand side of figure 1.

Figure 3. 3-D Graphical User Interface of Network Analysis Tool Developed in this paper.

Figure 4. ICMP Anomaly Example – The red perturbation was a series of 6,032 ICMP packets sent in a span of ten seconds.