Editors: Marcus Sachs, marcus.sachs@verizon.com
David Ahmad, drma@mac.com

Degradation and Subversion through Subsystem Attacks

Daniel Bilar
University of New Orleans

Say you wanted to harm your neighbor Vincent. You could go over to Vincent's home and punch him in the nose. You could be more sly, and poison his beer and wait till he harms himself by taking a sip. You could be even *more* sly and poison the hops of the beer brewing company. Though the latter two examples sound farfetched, these scenarios—attacking something or someone indirectly—have occurred on a larger scale in real life. I call these indirect attacks *nth order attacks* against end systems.

One way of defining a system is to view it as a whole that functions by virtue of the relationships between constitutive components. These components—I'll also call them ancillary systems—include control mechanisms, fault detection and recovery, energy/data flow, economic viability, human usability, data processing/structures, graceful startup/shutdown, and reputation management. Such ancillary systems can be embedded in or encompass an end system and can in turn be composed of and be influenced by other ancillary systems. A so-called *n*th order attack degrades, disables, or subverts a system by targeting one or more of its ancillary systems. The *n* stands for the degree of relation: 0th order targets the end system, first order targets an ancillary system of the end system, and so on.

Let's apply this view to a network intrusion detection system (NIDS). One ancillary system, the control subsystem, negotiates the data and instruction interplay between sensors, analysis system, database, and decision engine. Another ancillary system, the visualization subsystem, displays the events and possible remediation options. Drilling down, the visualization ancillary system has its own ancillary systems: human operators field a vision subsystem subject to certain parameters, among them a certain percentage of color-blindness and limited angular resolution. A human operator's control system is comprised of reasoning strength and limitations (for example, cognitive dissonance), as well as physiological mechanisms (hypothalamic hormone secretions that regulate sleep and hunger, for instance). The NIDS is embedded in a business model that governs aspects of its design, implementation, and activity (such as profit model and distribution channels). This business model is itself embedded in an economic environment, such as a free market economy, which influences its setup (tax codes and corporate structure, for instance).

How would you go about perpetuating an *n*th order attack against a NIDS? A first-order attack can target the visualization subsystem with noise events, overwhelming the visualization granularity. Why would this work? Attacks work because they violate assumptions. Any finite system—electronic or otherwise—must by design incorporate implicit and explicit assumptions into its structure, functionality, and language. Furthermore, systems are formulated with so-called "expected" or typical cases in mind, and assumptions reflect these expected use cases—a man-in-the-middle attack violates the assumption that you're talking to the party you expected; a buffer-overflow attack violates an explicit resource assumption; BGP routing and DNS case poisoning attacks violate implicit trust assumptions of non-malicious open architecture participants. I'll explain the importance of assumptions and expected cases with a model called highly optimized tolerance (HOT).

Highly Optimized Tolerance

HOT is a generative mechanism that seeks to explain the structure, static/dynamic attributes, and resiliency of interconnected systems.[1] Originally proposed to account for so-called "power law" distributions in natural and engineered systems, researchers have used the model to study forest ecosystems, router network robustness, Internet traffic, and power and immune systems. By emphasizing evolved and engineered complexity through feedback, trade-offs between objective functions and resource constraints in a probabilistic environment, HOT can capture a majority of real-life sys-

tems. Designs generated by HOT models generally perform well (as measured by throughput in router networks, for instance). In addition, they're robust towards designed-for uncertainties (so-called "average" cases) and hypersensitive to unanticipated perturbations ("rare" cases).

Let's tie the concepts of *n*th order and HOT together with an example of a 0th order attack. The equation system below is the formulation of a probability, loss, resource optimization problem (this PLR can be interpreted as a generalization of Shannon source coding for data compression, yielding the Shannon-Kolmogorov entropy for the objective function J[2]).

$$\min J \tag{1}$$

Subject to

$$\sum r_i \leq R \tag{2}$$

where

$$J = \sum p_i l_i \tag{3}$$

$$l_i = f(r_i) \tag{4}$$

$$1 \leq i \leq M \tag{5}$$

We have a set of M events (Equation 5) occurring independently and equally distributed with probability p_i incurring loss l_i (Equation 3), the sum product of which is the objective function to be minimized (Equation 1). Resources r_i are hedged against losses l_i, with normalizing $f(r_i) = -\log r_i$ (Equation 4) subject to resource bounds R (Equation 2). Now, I can map the resources and the loss to elements of a C program, then subject it to a 0th order attack (a buffer overflow):

```
int provePequalsNP()
{
 /* Next paper .. */
}
int bof()
{
 char buffer[8]; /* an 8
byte char buffer */
 strcpy(buffer, gets());
/*get input from user*/
 /* may not return if
buffer overflowed */
 return 42;
}

int main(int argc, char
**argv)
{
 bof(); /*call bof()
function*/
 /* execution may never
reach next function
because of overflow*/
 provePequalsNP();
 return 1000000; /*exit
with Clay prize*/
}
```

The probabilistic environment is the user, who is asked for input in `gets()`, representing the event. In the C code, the human designer specifies an 8-byte buffer `char buffer[8]` and the compiler allocates the minimum buffer needed for 8 bytes (resource r). Hence, the constrained resource r is the variable `buffer`. The loss associated with the user input event is really a step function: as long as the user satisfies the designer's assumption, the loss is just the "normal" loss incurred through proper continuation of control flow. As long as user input is ≤ 8 bytes, the resource r is minimally sufficient to ensure normal control-flow continuation. If, however, the user decides to input "Honorificabilitudinitatibus" (implicitly assumed to be an unlikely/impossible event by the human designer in the code), the loss function takes a huge step jump: a catastrophic failure ensues because `strcpy(buffer,gets())` overflows `buffer`. The improbable event breaches the resource hedge and the process crashes.

How did this vulnerability come about? I think two distinct HOT processes had a hand in allocating the breached resource. The first mechanism inducing a cost-optimized, resource-constrained executable program is the human programmer. As we all know,

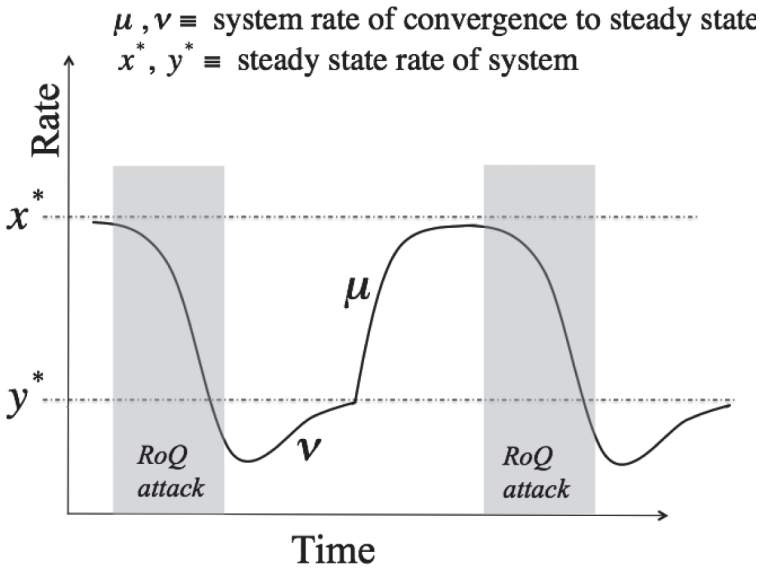

Figure 1. Oscillation between over- and underload conditions. Reduction of quality (RoQ) attacks force the adaptation mechanism with malicious traffic into dropping from a high system steady state rate (x^*) into a lower system steady state (y^*).

programmers juggle conflicting objective functions and resource constraints: the system's evolvability versus specificity, functionality versus code size, source readability versus development time, debugging time versus time to market. The second mechanism is the compiler. Cost functions here are memory footprint, execution cycles, and power consumption minimization, whereas the constraints typically involve register and cache line allocation, opcode sequence selection, pipelines, and arithmetic logic unit and floating point unit utilization.

nth Order Attacks on End Systems

More real-life illustrative attack examples exist—their salient characteristic lies in the targeting of ancillary systems to degrade or subvert respective end systems.

Protocols

Reduction of quality (RoQ) attacks constitute a first-order degradation attack,[3] which targets adaptation mechanisms used in network protocols. Non-denial-of-service, low-bandwidth traffic, maliciously optimized against the admission controllers and load balancers, forces the adaptive mechanism to oscillate between overload and underload conditions (Figure 1). The RoQ attack's δ requests per second for burst time t (grey shaded) repeated over period T constitutes the rare event that the adaptation system wasn't expected to handle efficiently. The adaptation mechanism—as a HOT process designed for common perturbations, but fragile toward rare events—finds its assumptions designed for normal traffic violated.

P2P Networks

RoQ attacks can be mounted against distributed hash tables used for efficient routing in structured P2P networks through join/leave collusions and bogus peer newcomer notifications.[4]

Power Grid

Load balancing in electricity grids relies on accurate state estimation. Data integrity attacks on a chosen subset of sensors make these estimates unreliable, which could push such feedback systems into an unstable state.[5]

Democracy

Voting systems assume honest participants vote their actual preference. In elections with more than two candidates, the system can be undermined by strategic voting, targeting the ranking process subsystem.[6]

Trusted Code

A second-order control-flow subversion attack termed return-oriented programming (ROP) has gained some notoriety. Its mechanism can induce innocuous code to perform malicious computations.[7] ROP vitiates the need for foreign code injection; as such, it renders security controls such as W $\oplus$ X obsolete. Detection schemes (shadow stacks for instance) exist, but are bypassed by new ROP implementations.

Financial Exchange

The semi-strong Efficient Market Hypothesis—a foundational assumption of financial markets—asserts that markets' prices assimilate past and present information near instantaneously. So-called statistical arbitrage algorithms, however, have been able to systematically generate profits over buy-and-hold strategies for 30 years, suggesting at least short-term exploitable inefficiencies. The advent of high-frequency trading infrastructures (physically collocated, hence low latency) gave rise to trading approaches targeting the EMH and its subsystems to the detriment of other market participants. So-called "Immediate or Cancel" price discovery algorithms used by automated market makers find the buy side's hidden limit order, forcing longer-term (predominantly institutional) investors to pay higher prices. Some market centers grant collocated (and thus latency-privileged) participants' algorithms a peek at future order data, enabling trading opportunities that would be illegal if humans were involved.[8] These and other predatory algorithms target in effect (and sometimes in intent) market price stability and transparency. As such, they constitute first- and second-order degradation and subversion attacks against the market.

My discussion of *n*th order attacks isn't merely of technical interest. It goes also to the heart of how conflicts between open societies and their enemies are waged: trust subsystems. Trust helps lower tangible and intangible transaction costs between individuals, corporations, and the state. Members of "high-trust" societies like the United States leverage trust beyond family ties to form efficient civic and economic organizations.[9] Because trust permeates every facet of open societies, it's a very easy assumption for malicious actors to violate. This realization wasn't lost on Jihadi terrorists articulating a 2nd order degradation attack strategy against open societies:

> [O]ur war with America is fundamentally different, for the first priority is defeating it economically [..] Any operation targeting an area of infrastructure in a new country that does not have a history of countering these operations is considered as bleeding (exhausting) to the greater enemy America and the targeted nation itself. It is so because these nations will be required to protect all similar potential targets which results in economic exhaustion (bleeding)... For example,

> if a hotel that caters to western tourists in Indonesia is targeted, the enemy will be required to protect all hotels that cater to western tourists in all countries which may become a target of similar attacks. You can say the same thing about living residences, economic establishments, embassies [..][10]

I will elaborate on these examples and discuss defenses in future columns. □

References

1. J. Carlson and J. Doyle, "Highly Optimized Tolerance: Robustness and Design in Complex Systems," *Physical Rev. Letters*, vol. 84, no. 11, 2000, pp. 2529--2532
2. J. Doyle and J. Carlson, "Power Laws, Highly Optimized Tolerance, and Generalized Source Coding," *Physical Rev. Letters*, vol. 84, no. 24, 2000, pp. 5656–5659
3. M. Guirguis and A. Bestavros, "Reduction of Quality (RoQ) Attacks on Internet End-Systems," *Proc. IEEE INFOCOM*, 2005, pp. 1362--1372
4. H. Yanxiang et al., "Reduction of Quality (RoQ) Attacks on Structured Peer-to-Peer Networks," *IEEE Int'l Parallel and Distributed Processing Symp.* (IPDPS 09), IEEE Press, 2009, pp. 1–9
5. Y. Liu, M.K. Reiter, and P. Ning, "False Data Injection Attacks against State Estimation in Electric Power Grids," *ACM Conf. on Computer and Communications Security* (CCS 09), ACM Press, 2009, pp. 21–32.
6. W. Poundstone, *Gaming the Vote: Why Elections Aren't Fair*, Hill and Wang, 2008
7. R. Roemer et al., "Return-Oriented Programming: Systems, Languages, and Applications," 2009, cseweb.ucsd.edu/~hovav/dist/rop.pdf.
8. H. Mittal, "Are You Playing in a Toxic Dark Pool ?," *J. Trading*, vol. 3, no. 3, 2008, pp. 20–33.
9. F. Fukuyama, *Trust: The Social Virtues and the Creation of Prosperity*. Free Press, 1996.
10. G. Ackerman and J. Tamsett, eds, *Jihadists and Weapons of Mass Destruction*, CRC Press, 2009, pp. 89–90.

Daniel Bilar *is an assistant professor of computer science at the University of New Orleans. He has degrees in Computer Science (Brown), Operations Research (Cornell) and Engineering Sciences (Dartmouth). //please indicate which of these is your highest degree (PhD?)// As a founding member of Dartmouth's Institute for Security, Technology, and Society (ists.dartmouth.edu), has conducted critical infrastructure protection research for the US Department of Justice and US Department of Homeland Security. Current research areas include detection and containment of highly evolved malware and compositional risk analysis and management of networks. Contact him at daniel@cs.uno.edu.*