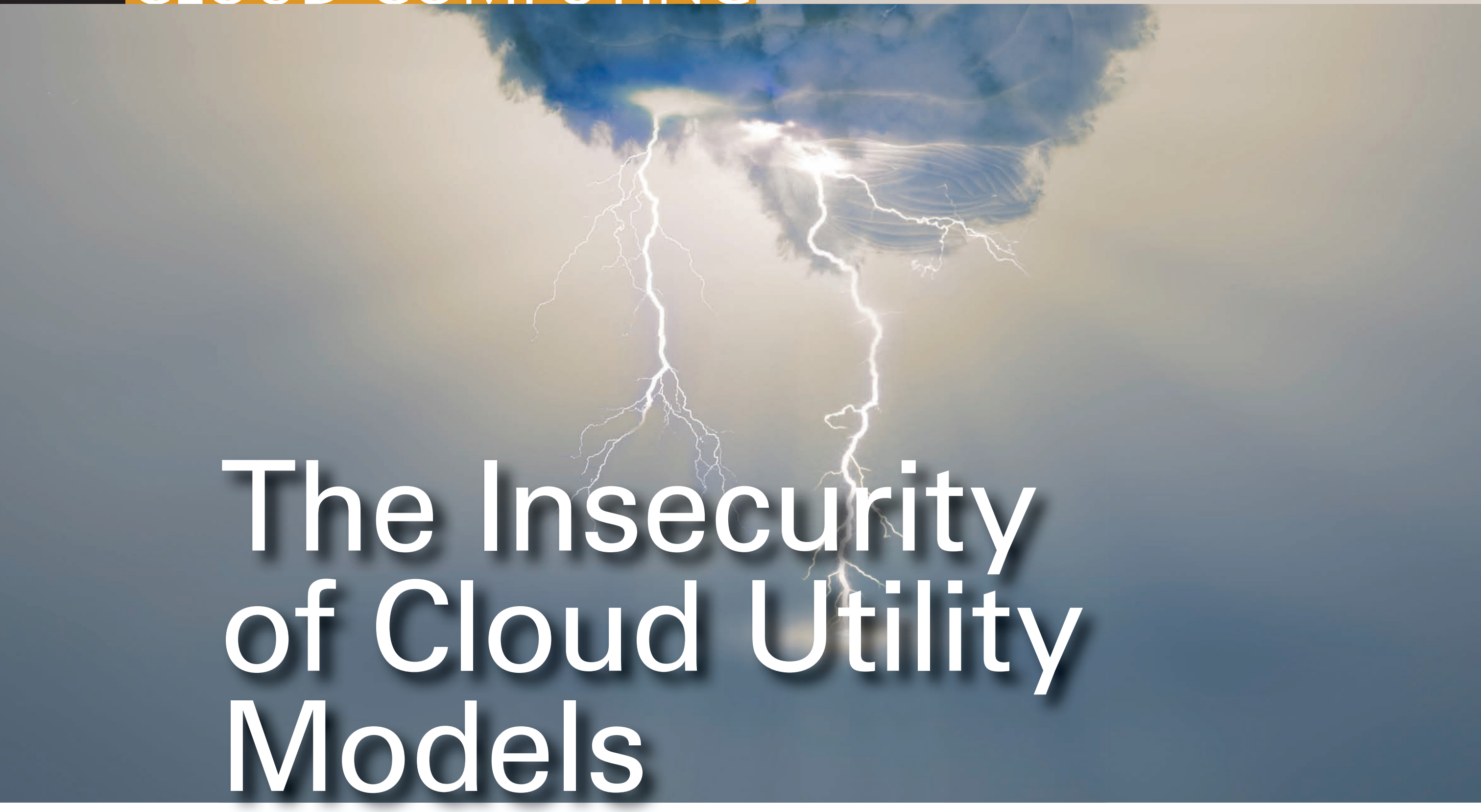

The Insecurity of Cloud Utility Models

Joseph Idziorek, Mark F. Tannian, and Doug Jacobson, *Iowa State University*

Cloud-based services are vulnerable to attacks that seek to exploit the pay-as-you-go pricing model. A botnet could perform fraudulent resource consumption (FRC) by consuming the bandwidth of Web-based services, thereby increasing the cloud consumer's financial burden.

A key feature that has led to the early adoption of public cloud computing is the utility pricing model, which governs the cost of computing resources consumed. Similar to public utilities, such as gas and electricity, cloud consumers only pay for the resources (storage, bandwidth, and computer hours) they consume and for the time they use such resources. In accordance with the terms of agreement of the cloud service provider (CSP), cloud consumers are responsible for all computational costs incurred in their leased compute environments, regardless of whether the resources were consumed in good faith.

Common use cases for corporations that have adopted public cloud computing include website and Web application hosting and e-commerce. Like any Internet-facing presence, these cloud-based services are vulnerable to distributed denial-of-service (DDoS) attacks. Such attacks are well known, and the associated risks have been well researched. Here, we explore a more subtle attack on Web-based services hosted in the cloud. Given the pay-as-you-go pricing, cloud-hosted Web services are vulnerable to attacks that seek to exploit this model. An attacker (for example, a botnet) can perform a *fraudulent resource consumption* (FRC) attack by consuming the metered bandwidth of Web-based services, increasing the cloud consumer's financial burden.[1,2]

In the scenario in Figure 1, a botnet—comprising potentially thousands of bot clients—is consuming Web resources hosted in the cloud by

Published by the IEEE Computer Society

1520-9202/13/$31.00 © 2013 IEEE

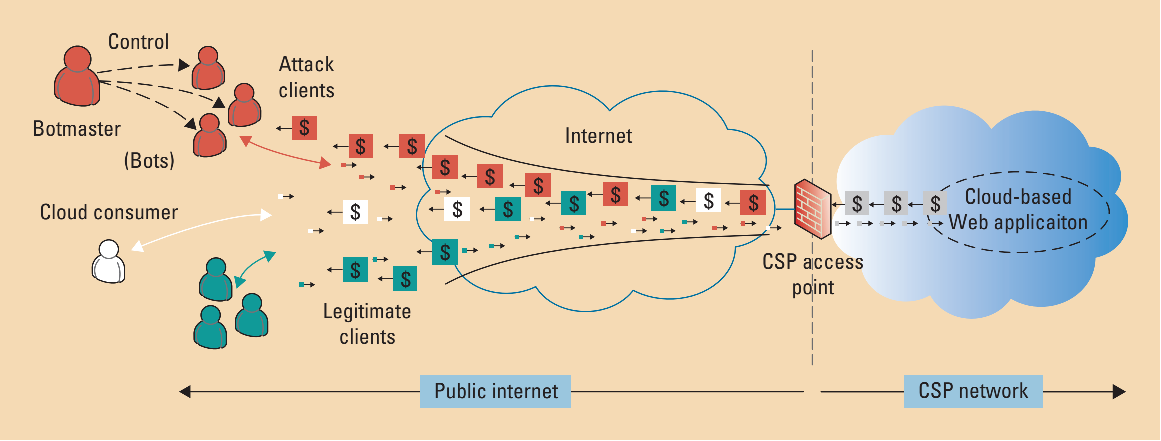

Figure 1. **A cloud network-attack diagram. Botnets can exploit the cloud utility model to perform *fraudulent resource consumption* (FRC), making consumers incur unexpected costs from dishonest use.**

mimicking legitimate client behavior. To the cloud-based Web application, the intention of incoming requests is either unknown or not considered, so each request is serviced with a reply, resulting in a fractional cost for the cloud consumer. Because this vulnerability, up until now, hasn't been largely discussed, determining this threat's overall effect on the cloud community is difficult. Rather, we focus here on describing the vulnerability to increase awareness, analyzing the risk for an individual cloud consumer and discussing methods for FRC prevention, detection, attribution, and mitigation.

The Utility Model

The utility model is attractive to a cloud consumer because the low entry cost removes the burden of major capital expenses. However, although convenient, the utility model isn't without its risks—the financial liability for resources consumed is unlimited. CSPs, such as Amazon EC2 and Rackspace, charge US$0.12 per Gbyte (up to 40 Tbytes) and $0.18 per Gbyte, respectively, for outbound data transfers.[3,4]

As Figure 1 shows, the cloud consumer (the victim) incurs a cost each time a cloud application (the attack target) services a reply. A high volume of requests can be costly. Malicious use is even more burdensome, because the additional run-up in expenses has no associated business value. As it stands today, CSPs don't monitor cloud consumers' applications, so it's up to the cloud consumer to prevent, monitor, and respond to such fraudulent behavior.

Fraudulent Resource Consumption

To better understand the FRC attack, consider the time-series visualization of a Web server log shown in Figure 2.[1,2] The y-axis depicts the number of requests per second, and as the x-axis shows, the time series covers a two-week period. As is common, the modeled Web server capacity is sufficiently over-provisioned—this represents a conservative estimate, given the capacity of CSP Web servers. Superimposed on top of normal Web activity are serviced requests from an FRC attack.

As Figure 2 shows, initial attack intensity beyond normal activity is in the "nuisance activity" region, because the resultant costs are insignificant to the cloud consumer. However, as malicious activity intensifies beyond this region, the malicious costs to the cloud consumer start to become a matter of concern; this transition point is labeled J1. Malicious activity that exceeds J1 enters into the FRC attack region. Within this region, bounded by J1 and J2, an FRC attack doesn't significantly degrade the Web server's quality of service (QoS).

If the attack intensity increases above J2, the request volume will reach a point at which the Web server QoS starts to significantly degrade. At this point, current application-layer DDoS detection and mitigation schemes are effective.[5] An objective of FRC attack mitigation research is to improve detection sensitivity that will push J2 closer to J1, thus narrowing the FRC attack region by detecting attacks that are legitimate transactions but differ in the requestors' intent.

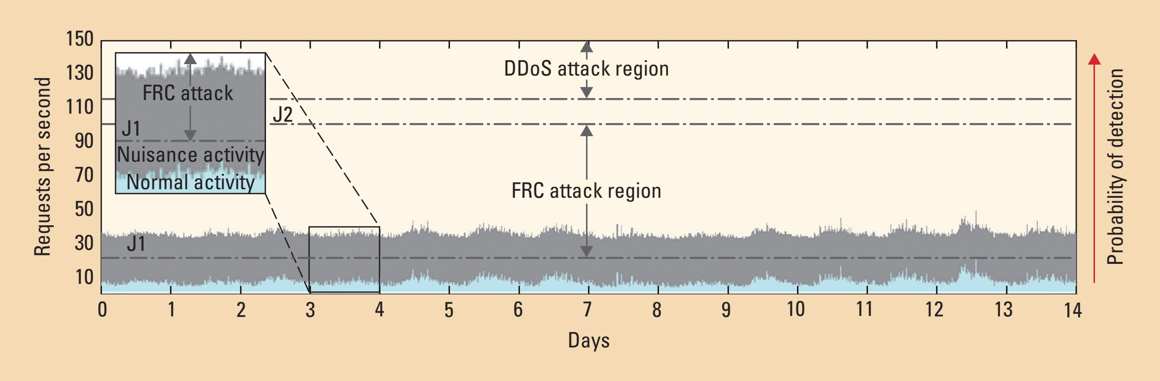

Figure 2. Malicious-requests behavior. The initial attack intensity (labeled J1) results in insignificant costs for the cloud consumer. However, as malicious activity intensifies beyond this "nuisance activity" region, the cost to the consumer starts to become a matter of concern. Yet distributed denial-of-service detection schemes aren't effective at this lower intensity level (below J2).

As shown on the right side of Figure 2, the probability of detecting an FRC attack increases as the attack intensity increases. Although nothing prevents an attacker from exploiting the utility model with an attack intensity in the DDoS attack region, such a blatant action carries a higher risk of detection and ultimately mitigation. Depending on the attack objectives and the cost to the attacker, a modest request intensity within the FRC attack region over an extended duration of time has a higher chance of success, because this is considerably more difficult for a victim to mitigate.

Faced with such an attack, current DDoS mitigation schemes, firewalls, and intrusion prevention and detection systems would be rendered ineffective, because individual fraudulent requests are protocol compliant and attack rates don't degrade the Web server QoS. As a result, and given the utility pricing model, the potential for an FRC attack fundamentally changes requirements of Web-based anomaly detection for the cloud.

Figure 3 depicts an FRC attack as a slow-and-low assault or "death by a thousand requests." Unlike short-lived DDoS attacks, the duration of an FRC attack could last weeks or months if not detected. Because resources maliciously consumed are additive to that of normal traffic, the aggregate of legitimate and malicious resource use is reflected in a cloud consumer's monthly bill.

Availability in the context of this discussion isn't a binary measure in which the system is nearly incapacitated at the time of the attack. The technical infrastructure of a website hosted in a CSP environment will have no trouble functioning while an FRC attack is underway. Instead, availability is a long-term consideration defined as the cloud consumer's ability to withstand the financial consequences of an FRC attack over a prolonged time period.

FRC Risk

Adopting the public cloud model brings with it new and old security risks. Here, we focus on the risk introduced by the utility pricing model by discussing the likelihood and effects of an FRC attack.

The likelihood of a cloud consumer falling victim to an FRC attack depends on the attacker's skill level, computing capacity, and motivation as well as his or her ability to exploit the utility pricing model. This pricing vulnerability is literally hiding in plain sight, because CSPs openly publish their pricing metrics. From a technical standpoint, all that's necessary for an attacker to exploit this vulnerability is to make standard requests for Web content that the cloud consumer makes publicly available. Although a large botnet is the worst-case threat source, conceivably any Internet-connected device could perform an FRC attack with a Perl script making HTTP GET requests or using the Low-Orbit Ion Cannon—an open-source tool that has fueled recent DDoS attacks.[6]

As evidenced by the growing number, capacity, and sophistication of both botnets and DDoS attacks, the worst-case threat sources undoubtedly possess the skills and resources to mount a sustained and effective FRC attack. The only real factor preventing an FRC attack is a lack of motivation. Yet similar to those who orchestrate

DDoS attacks, the motive of an FRC attacker could range from ego and hacktivism to monetary gain, extortion, revenge, competitive advantage, or economic espionage.[7] If recent history is any guide, those who control botnets could perform an FRC attack to promote a political agenda or support an ideological viewpoint.

For the victim, the direct monetary effect of an FRC attack is a function of the average request intensity and attack duration. To enumerate one end of the extreme, a week-long DDoS attack launched from a 250,000 node botnet in 2011 peaked at 45 Gbps.[8] If the aforementioned attack peak was sustained on a cloud instance at $0.12/Gbyte, the resultant costs would have been $0.675 per second—which adds up to $411,264 per week.

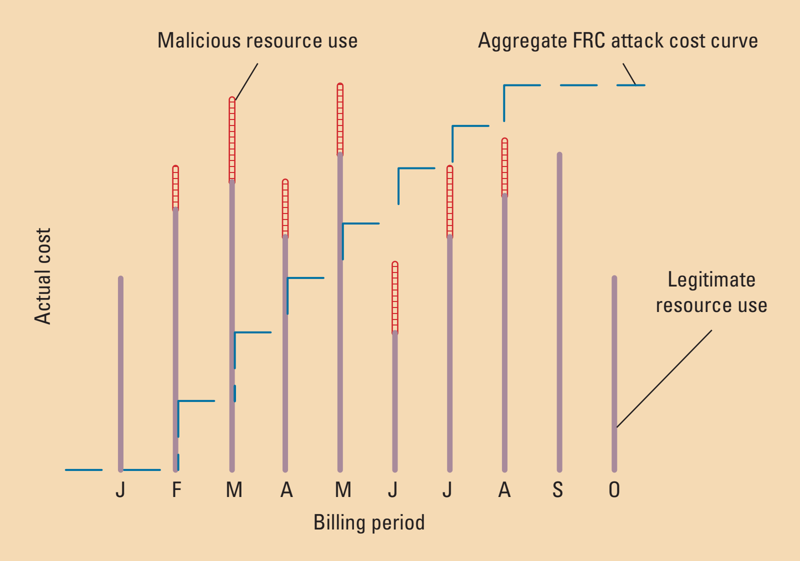

Figure 3. Aggregation of an FRC attack—a slow-and-low assault. Unlike short-lived DDoS attacks, an FRC attack could last weeks or months if not detected.

On the other end of the FRC attack region, consider the website modeled in Figure 2. At an average normal request rate of three requests per second, a 250,000-node botnet could double the data usage costs if each bot client generated just two requests per day. Clearly, given the capacity of modern-day networks and computers, the bot clients in this example could significantly increase their daily request quota and multiply the attack cost by orders of magnitude. However, once a bot client's usage footprint eclipses the expected behavior of legitimate clients, the risk of being identified as malicious greatly increases.

Defending against an FRC Attack

Defending against an FRC attack is a significant challenge to the cloud consumer, owing to the atypical and unassuming nature of the attack. As is the case with most attack risks, the cloud consumer has four primary objectives: prevention, detection, attribution, and mitigation.

Prevention

A common way to prevent the exploitation of a vulnerability is to download and apply a patch for it. However, in the context of this discussion, the "bug" isn't a software defect but a common business model deployed by CSPs. Until this vulnerability is actually exploited, the cloud business model isn't likely to change. So in lieu of a patch for this vulnerability, there are several, albeit limited, prevention options.

The use of authentication on a target website would significantly reduce the amount of exploitable resources, but we don't consider it here because we assume the cloud consumer wants to host public content. Similarly, graphical puzzles (Captcha tests) could be used as a preemptive solution to differentiate humans and zombie computers. However, the use of such a test could be detrimental to the overall goals of a public-facing website, because these types of tests will result in a certain percentage of legitimate clients being unable or unwilling to solve such puzzles.

Another option would be for the cloud consumer to work with application and content developers to minimize the resource footprint of common or average requests. Limiting the impact of client requests increases FRC attacker costs and risk of detection. Unfortunately, without a utility model patch, these controls won't thwart a motivated attacker. So with limited prevention capability, the next line of defense is detection.

Detection

FRC detection aims to identify malicious traffic consumption. Because an FRC attack is subtle, previous application-layer DDoS solutions that focus on high request intensities aren't suitable.[9]

Instead, initial FRC-detection approaches focus on behavioral metrics derived from Web server log files that seek to profile the aggregate webpage request choices of a website's client base.[2] Three measures—the Spearman, Overlap, and Zipf metrics—respectively characterize the accuracy, completeness, and relative proportionality of ranked requests between two adjacent windows of observed logs (for example, two three-day windows).[2] Together, these three metrics provide consistent measures with which to describe normal behavior and perform anomaly detection.

However, for the sake of brevity, we don't present empirical results here. The conclusion stemming from this work is that an attacker, without knowledge of the training dataset (historical Web server log), has a difficult time requesting an impactful volume of Web documents while adhering to the structure of normal traffic.

This attribution methodology operates under the condition that all clients are innocent until their usage footprint proves otherwise.

Thus our proposed methodology,[2] which focuses on characterizing aggregate Web traffic, is effective for detecting even minor increases in fraudulent Web activity, well before the resultant costs are harmful.

The most practical detection approach is the classic "review your bills" approach. Reviewing bills over time to determine if they're within an expected range can help expose an FRC attack. Log analyzers might also help identify outlier application usage, triggering an investigation of suspicious clients. A casual inspection, however, won't catch a savvy FRC attacker.

Attribution

Attribution in this context is the ability to accurately differentiate legitimate clients from FRC attack clients. Like the previously discussed DDoS detection solutions, current attribution solutions are geared toward detecting malicious clients that consume a significant volume of requests in a very short time. Previous work has focused on scrutinizing the increased interrequest (the time between successive Web document requests) or intersession (the time between Web browsing sessions) arrival request rates of malicious clients in comparison to the rate profile for normal users.[10] Again, it's contrary to FRC attack objectives for a single attack client to behave in a fashion similar to one participating in a DDoS attack.

The challenge in this research area will be to minimize the number of falsely identified legitimate clients while decreasing the impact of fraudulent clients. Recent research indicates that normal client behavior can be characterized by client actions such as request volume per client, Web documents requested, and Web session parameters (for example, requests per session and number of sessions).[11] If attack clients that aren't privy to normal usage activity exceed a set threshold on these characteristics, they're flagged as malicious.

This attribution methodology aims to be transparent to clients, and it operates under the condition that all clients are innocent until their usage footprint proves otherwise. Limiting the impact of individual clients reduces the overall risk of an FRC attack. It's important to note that this methodology is not rate-based; rather, it's sensitive to the accumulated requests an attacker invokes. Therefore, the choices an attacker makes could allow a malicious client to be deemed anomalous after it invokes a minimal number of requests.

Mitigation

Reactive solutions rely on accurate detection and attribution. We must consider the potential for legitimate clients being errantly classified as malicious. As a result, approaches like blacklisting first-time offenders might prove heavy-handed. Less absolute mitigation strategies include imposing a back-off timeout to anomalous clients in which requests from an IP address aren't all serviced. Similarly, suspicious clients could also be served a graphical puzzle to prove that the client is indeed a human.

These reactive approaches are available today, and each has its own tradeoffs. However, with limited detection and attribution solutions available, the deployment and maintenance of such solutions will be challenging.

Letting any client with access to the Internet consume resources that are in turn metered and billed exposes the cloud consumer to a risk that's only mitigated by time, detection, and accountability. Until recently, this vulnerability has been neglected. Unless utility models are restructured to remove the vulnerability of an FRC attack, research in detection and attribution is necessary to ensure the long-term sustainability of cloud consumers and remove one more impediment that could dissuade organizations from adopting public cloud computing.

To the best of our knowledge, there have been no known public acknowledgements of an FRC attack occurring on the public cloud. However, the absence of such knowledge doesn't confirm that the utility model vulnerability hasn't or won't be exploited. Back in the early 1990s, Internet-facing firewalls were new and thought to be sufficient to secure a connected enterprise. In reality, attacks were occurring, as intrusion-detection systems soon pointed out. Perhaps the utility model has been exploited and, as an IT community, we're presently ill-equipped to detect its presence or identify its culprits. IT

References

1. J. Idziorek and M. Tannian, "Exploiting Cloud Utility Models for Profit and Ruin," *Proc. 2011 IEEE 4th Int'l Conf. Cloud Computing* (Cloud 11), IEEE, 2011, pp. 33–40.
2. J. Idziorek, M. Tannian, and D. Jacobson, "Detecting Fraudulent Use of Cloud Resources," *Proc. 3rd ACM Workshop on Cloud Computing Security Workshop* (CCSW 11), ACM, 2011, pp. 61–72.
3. "Amazon EC2 Pricing," Amazon Web Services, 2012; http://aws.amazon.com/ec2/pricing.
4. "Cloud Servers Pricing," Rackspace Cloud Servers, 2012; www.rackspace.com/cloud/cloud_hosting_products/servers/pricing.
5. S. Kandula et al., "Botz-4-Sale: Surviving Organized DDoS Attacks that Mimic Flash Crowds," *Proc. 2nd Conf. Symp. Networked Systems Design & Implementation*, Usenix, 2005, pp. 287–300.
6. L. Page, "Join in the Wikileaks DDoS War from your iPhone or iPad," *The Register*, 10 Dec. 2010; www.theregister.co.uk/2010/12/10/loic_for_iphone.
7. G. Stonebumer, A. Goguen, and A. Feringa, "Risk Management Guide for Information Technology Systems," NIST Special Publication 800-30, July 2002.
8. L. Constantin, "Denial-of-Service Attack Are on the Rise, Anti-DDoS Vendors Report," IDG News Service; 7 Feb. 2012; www.pcworld.com/businesscenter/article/249438/denialofservice_attacks_are_on_the_rise_antiddos_vendors_report.html.
9. S. Wen et al., "Cald: Surviving Various Application-layer DDoS Attacks that Mimic Flash Crowd," *Proc. 2010 4th Int'l Conf. Network and System Security* (NSS 10), IEEE, 2010; pp. 247–254.
10. S. Ranjan et al., "DDoS-Shield: DDoS-Resilient Scheduling to Counter Application Layer Attacks," *IEEE/ACM Trans. Networking*, Feb. 2009, pp. 26–39.
11. J. Idziorek, M. Tannian, and D. Jacobson, "Attribution of Fraudulent Resource Consumption in the Cloud," *Proc. 2012 IEEE 5th Int'l Conf. Cloud Computing* (Cloud 12), IEEE, 2012, pp. 99–106.

Joseph Idziorek *is a PhD candidate in the Department of Computer and Electrical Engineering at Iowa State University. His research interests broadly include anomaly detection and more specifically the detection and attribution of FRC attacks on the cloud utility model. Idziorek received his BS in computer engineering from St. Cloud State University. Contact him at idziorek@iastate.edu.*

Mark F. Tannian *is a PhD candidate in the Department of Computer and Electrical Engineering at Iowa State University. His research interests include user-centered design and information security visualization in addition to cloud computing security. Tannian received his MS in electrical engineering from George Washington University. Contact him at mtannian@iastate.edu.*

Doug Jacobson *is a University Professor in the Department of Computer and Electrical Engineering at Iowa State University, where he serves as the director of the Information Assurance Center. His research interests include Internet-scale event and attack generation environments. Jacobson received his PHD in computer engineering from Iowa State University. Contact him at dougj@iastate.edu.*