Poster: LEADER (Low-Rate Denial-of-Service Attacks Defense)

Rajat Tandon, Haoda Wang, Nicolaas Weideman, Christophe Hauser, Jelena Mirkovic
Information Sciences Institute, University of Southern California
tandon, haodawa, nweidema, hauser, sunshine @isi.edu

***Abstract*—Low-rate denial-of-service(LRD) attacks are often hard to detect at the network level as they consume little bandwidth. Both the legitimate traffic and the attack traffic look alike. Moreover, the attack traffic often appears to comply with transport protocol, and application protocol semantics. It is the intricacies in the payloads and the dynamics of the attack traffic that induces denial-of-service on servers when processed by specific hardware and software.**

We introduce Leader, a hybrid approach for application-agnostic and attack-agnostic detection and mitigation of LRD attacks. Leader operates by learning normal patterns of network, application and system-level resources when processing legitimate external requests. It relies on a novel combination of runtime, system and network monitoring and offline binary program analysis.

I. Introduction

Low-rate denial-of-service attacks deny service at the device level (server, router, switch), and stay well below the network bandwidths limit. These attacks use a specific basic mechanism to deny service. They deplete some limited resource at the application, the operating system or the firmware of a device. This makes the device unable to process legitimate clients' traffic. Examples of LRD attacks are: (1) Connection depletion attacks (e.g., TCP SYN flood [3] or SIP flood [2], which deplete a connectiontable space for a given service by keeping many half-open connections, (2) Application/System-level depletion attacks ((e.g., ZIP bombs [6]), which deplete CPU through expensive and never-ending computation operations).

Leader protects the deploying server against every variant of LRD attack. The novel aspect of leader is it's hybrid detection approach combining network security mechanisms with OS and program-level aspects. Leader operates by learning, during baseline operation, a normal pattern of an applications and the devices use of system resources, when serving external requests. Leader detects attacks as cases of resource overload that impairs some services quality. It then runs diagnostics, compares the observed and the expected resource usage patterns, and checks for anomalies. This helps it diagnose the attack type, and select the best remediation actions. Another novel aspect of Leader lies in the structures it uses to capture sequences of resource-use events in a temporal and relational manner per each incoming service request. These sequences are known as connection life stages. They are built from multiple, complementary observations collected at the (1) network level, (2) OS level and (3) application level. The connection life stages are then clustered into typical resource-use patterns, or profiles for applications and for users. We use these profiles to detect anomalous use and characterize LRD attacks. We also design mitigation actions that remove attack traffic, or increase systems robustness to the specific attack with the aid of these profiles.

II. Methodology

Figure 1(a) shows a high-level view of the abstraction levels at which Leader operates. It also illustrates the trade-offs between accuracy of semantic reasoning on one side, and monitoring cost and delay on another side. Semantic reasoning is accurately understanding and attributing resource usage to a given application and network connection. Better accuracy implies more monitoring, which incurs higher cost and higher delay. We qualify as black box observation the process of characterizing the behavior of an application based on the sole observation of its inputs and outputs. OS-level instrumentation allows an observer to gain more insights about the applications semantics and program analysis offers the highest level of semantic reasoning.

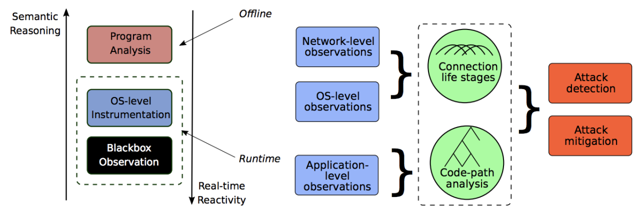
Figure 1(b) gives the system overview. LRD attacks usually involve one or several incoming service requests arriving at the server that are expensive or slow processing leading to resource depletion. This causes a state where the service or the global system is unable to gracefully handle new requests and denial of service occurs. The defense mechanisms in Leader comprises the following operational steps unified into the three main modules: (1) Behavior profiling, (2) Attack detection (and characterization), and (3) Attack mitigation.

A. *Behavior profiling*

LEADER relies on the collection of measures and statistics of system resources usage for successful attack detection. These measures are collected per each incoming service request and comprise observations at the network, system and application level. Leader captures the connection, client, application and whole-device profiles at all times. During normal operation, Leader performs profiling to learn the legitimate behavior of the connections, applications, clients and the device where it is deployed.

B. *Attack detection*

Another novel aspect of leader is that it uses a hybrid detection approach combining network security mechanisms



(a) Monitoring at different abstraction levels, with the associated trade-off.

(b) System overview.

Fig. 1. Monitoring and system overview: novelty of our approach lies in our connection life stages and code path abstractions, which are built from monitoring the system at network, OS and application levels.

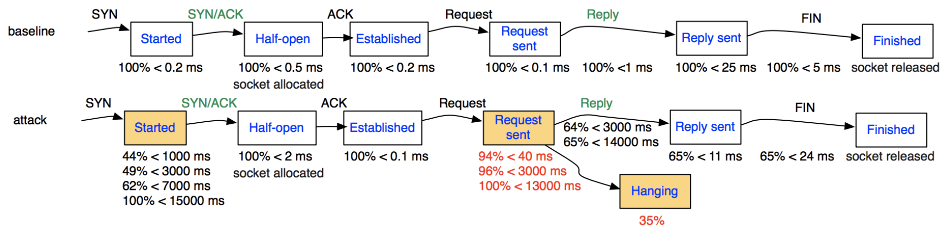


Fig. 2. Life-stage diagrams for Slowloris attack: highlighted items show anomalies.

with OS and program-level aspects. Our attack detection module compares instantaneous profiles to corresponding baseline profiles, with the goal of detecting evidence of troubled services that have low instantaneous percentages of successfully served incoming requests, compared to the historical (baseline) percentages of successfully served request.

C. Attack mitigation

Leader deploys a combination of attack mitigation approaches that includes (1) Derivation of the attack signature from attack connections (2) Costly connection termination, (3) Blacklisting of attack sources, (4) Dynamic resource replication, (5) Program patching and algorithm modification, and (6) Blacklisting of sources with anomalous profiles.

III. Preliminary Findings and Next Steps

We relied on on Emulab [4] to experiment with Slowloris [5], a common type of LDR attacks. We set up a static Web server, and used 10 legitimate and one attack client. The legitimate clients continuously requested the main Web page, using wget, each 200 ms. This created 50 requests per second at the server. The attack client opened 1,000 simultaneous attack connections and kept them open for as long as possible by sending never-ending headers on each. This had negative impact on legitimate clients, that had trouble establishing connection with the server. Figure 2 shows the life stage diagram for traffic in attack case, which includes both legitimate and attack connections, and compared it to the diagram in the baseline case. The highlighted stages differ between two cases and help us identify anomalous connections.

Currently, we use Systemtap [1] to build aggregate profiles of an application's/system's resource usage pattern for each connection at the system call level. We capture the time spent on each system call and the resources used by them such as memory usage, number of page faults and open file descriptors, cpu cycles and other thread/process level details. We do such profiling for both legitimate traffic as well as attack traffic. There are notable differences between the two. We will utilize these profiles, along with other sophistication, to detect attacks and mitigate them.

IV. Conclusion

LEADER leverages a novel combination of runtime monitoring and offline binary program analysis to protect a deploying server against LRD attacks and prevents external service requests from misusing system resources. During baseline operation, LEADER learns resource-usage patterns and detects and mitigates attacks by following an anomaly detection paradigm.

References

[1] Eigler, F. C., and Hat, R. Problem solving with systemtap. In *Proc. of the Ottawa Linux Symposium* (2006), pp. 261–268.

[2] Luo, M., Peng, T., and Leckie, C. Cpu-based dos attacks against sip servers. In *Network Operations and Management Symposium, 2008. NOMS 2008. IEEE* (2008), IEEE, pp. 41–48.

[3] Manna, M. E., and Amphawan, A. Review of syn-flooding attack detection mechanism. *arXiv preprint arXiv:1202.1761* (2012).

[4] White, B., Lepreau, J., Stoller, L., Ricci, R., Guruprasad, S., Newbold, M., Hibler, M., Barb, C., and Joglekar, A. An integrated experimental environment for distributed systems and networks. *ACM SIGOPS Operating Systems Review 36*, SI (2002), 255–270.

[5] Wikipedia. Slowloris. *https://en.wikipedia.org/wiki/slowloris_(computer_security)*.

[6] Wikipedia. Zip bomb. *https://en.wikipedia.org/wiki/Zip_bomb*.

LEADER: Low-Rate Denial-of-Service Attacks Defense

Rajat Tandon - Haoda Wang - Nicolaas Weideman - Christophe Hauser - Jelena Mirkovic

Information Sciences Institute, University of Southern California

Low-rate denial-of-service(LRD) attacks are often hard to detect at the network level as they consume little bandwidth. It is the intricacies in the payloads and the dynamics of the attack traffic that induces denial-of-service on servers when processed by specific hardware and software.
We introduce **Leader, a hybrid approach for application-agnostic and attack-agnostic detection and mitigation of LRD attacks.** Leader operates by **learning normal patterns of network, application and system-level resources** when processing legitimate external requests. It relies on a **novel combination of runtime, system and network monitoring and offline binary program analysis.**

Overview

Monitoring and system overview: **Novelty of our approach lies in our connection life stages and code path abstractions, which are built from monitoring the system at network, OS and application levels.** Figure 1 shows a high-level view of the abstraction levels at which Leader operates. It also illustrates the trade-offs between accuracy of semantic reasoning and, monitoring cost and delay. OS-level instrumentation allows an observer to gain more insights about the applications semantics and program analysis offers the highest level of semantic reasoning. Figure 2 gives the system overview. LRD attacks usually involve several incoming service requests arriving at the server that are expensive/slow processing leading to resource depletion.

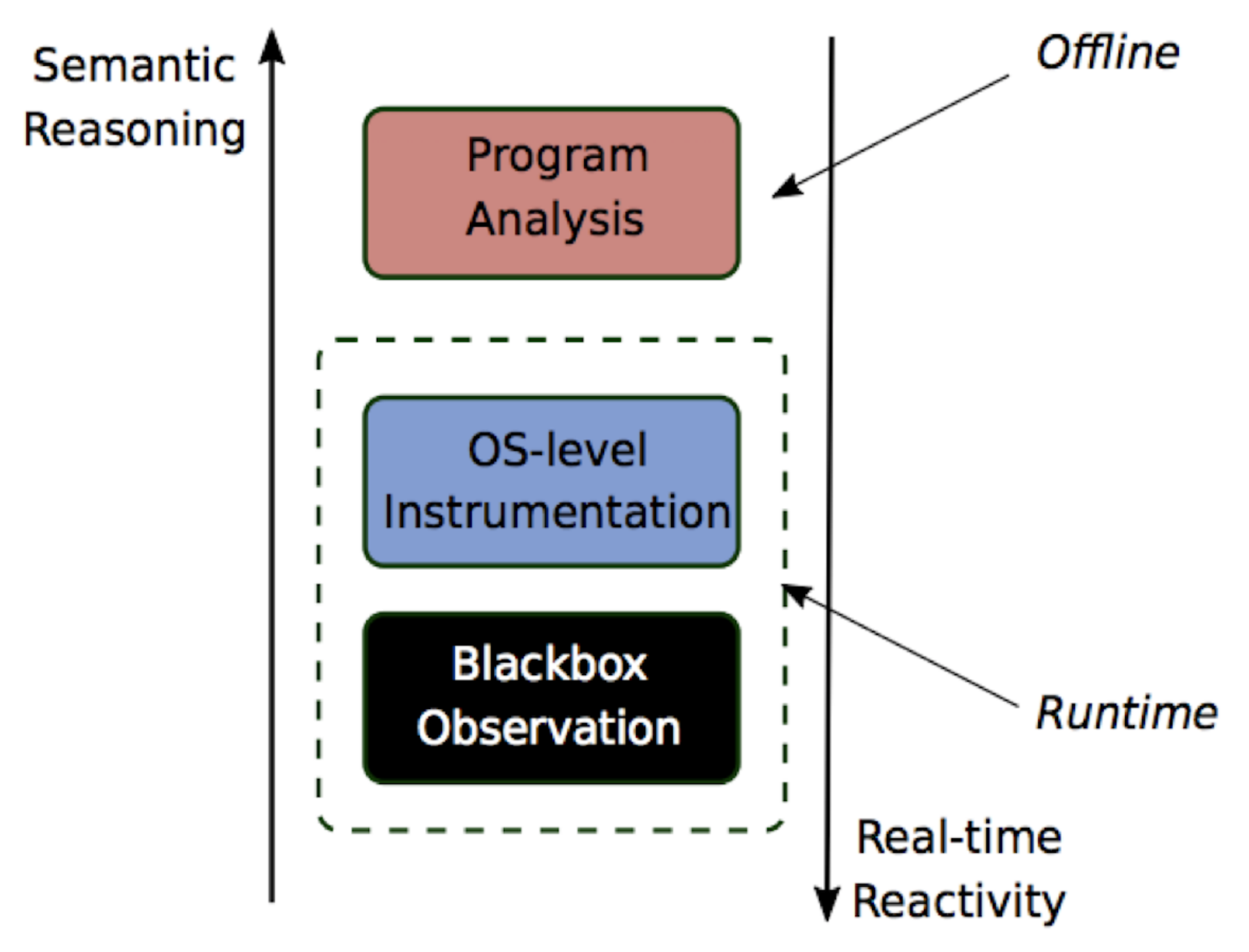


Figure 1: Monitoring at different abstraction levels, with the associated trade-off.

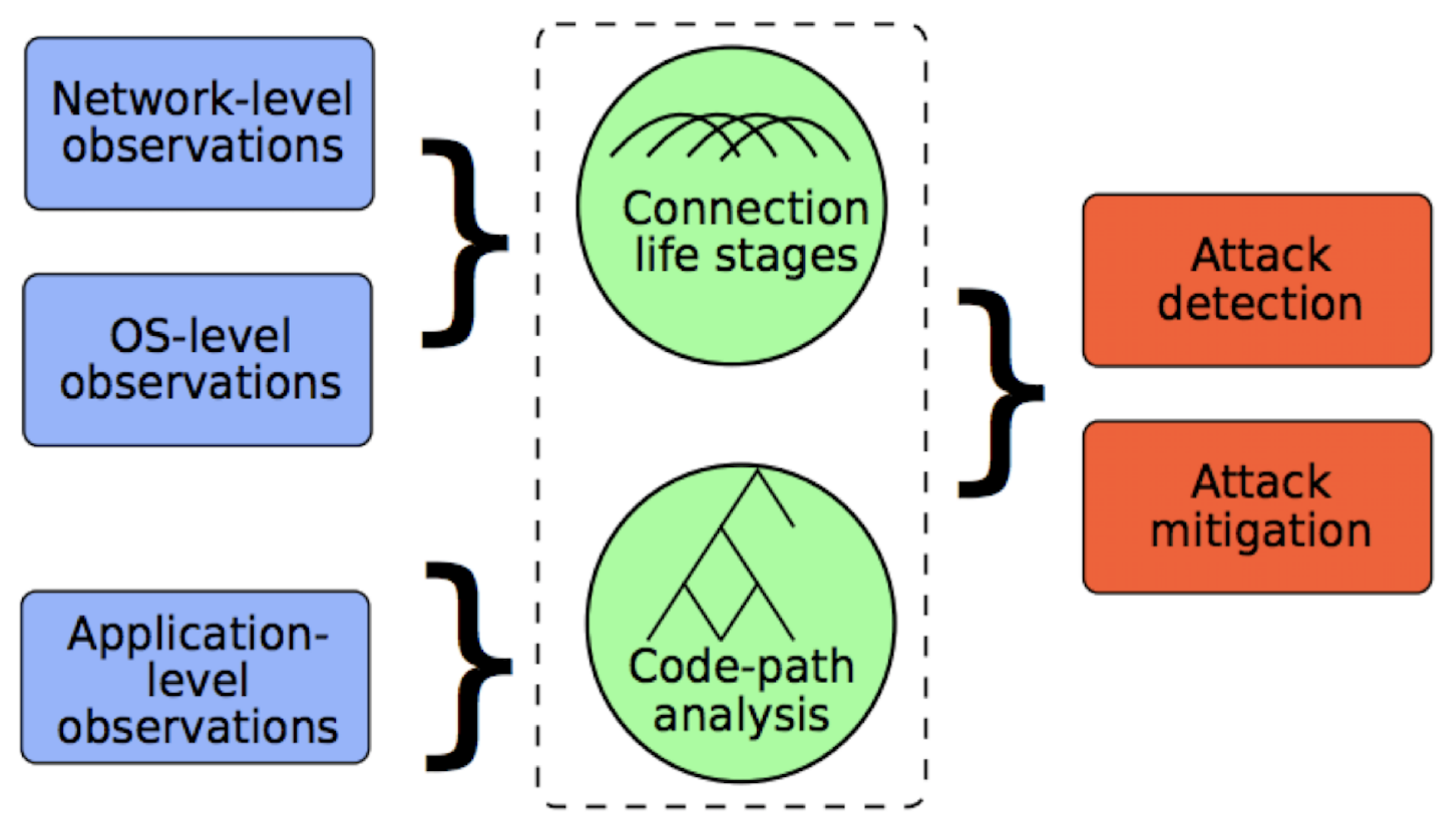


Figure 2: System overview.

Another novel aspect of Leader lies in the structures it uses to capture sequences of resource-use events in a temporal and relational manner per each incoming service request. These sequences are known as **connection life stages**. We relied on on Emulab to experiment with Slowloris, a common type of low-rate denial-of-service attack. We set up a static Web server, and used 10 legitimate and one attack client. The legitimate clients continuously requested the main Web page, using wget, each 200 ms. This created 50 requests per second at the server. The attack client opened 1,000 simultaneous attack connections and kept them open for as long as possible by sending never-ending headers on each. This had negative impact on legitimate clients, that had trouble establishing connection with the server. Figure 3 shows the life stage diagram for traffic in attack case, which includes both legitimate and attack connections, and compares it to the diagram in the baseline case.

Methodology

- **Behavior profiling**

LEADER relies on the collection of measures and statistics of system resources usage for successful attack detection. These measures are collected per each incoming service request and comprise observations at the network, system and application level. Leader captures the connection, client, application and whole-device profiles at all times. During normal operation, Leader performs profiling to learn the legitimate behavior of the connections, applications, clients and the device where it is deployed.

- **Attack detection (and characterization)**

Another novel aspect of leader is that it uses a hybrid detection approach combining network security mechanisms with OS and program-level aspects. Our attack detection module compares instantaneous profiles to corresponding baseline profiles, with the goal of detecting evidence of troubled services that have low instantaneous percentages of successfully served incoming requests, compared to the historical (baseline) percentages of successfully served request.

- **Attack mitigation**

Leader deploys a combination of attack mitigation approaches that includes (1) Derivation of the attack signature from attack connections (2) Costly connection termination, (3) Blacklisting of attack sources, (4) Dynamic resource replication, (5) Program patching and algorithm modification, and (6) Blacklisting of sources with anomalous profiles.

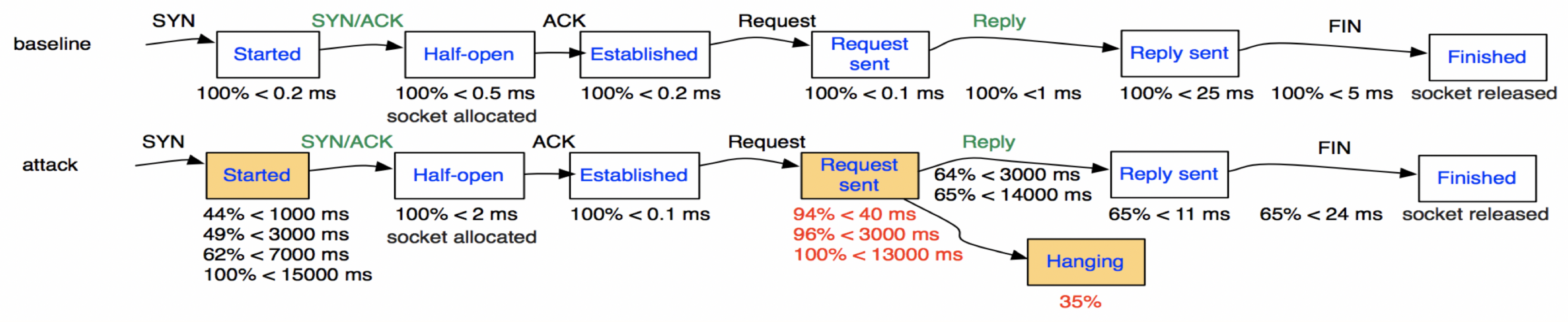


Figure 3: Life-stage diagrams for Slowloris attack: highlighted items show anomalies.

We are working on **Systemtap** to build aggregate profiles of an application's/system's resource usage pattern for each connection at the system call level. We capture the time spent on each system call and the resources used by them such as memory usage, number of page faults and open file descriptors, cpu cycles and other thread/process level details. We do such profiling for both legitimate traffic as well as attack traffic. There are notable differences between the two. We will utilize these profiles, along with other sophistication, to detect attacks and mitigate them.

Contact us at — tandon, haodawa, nweidema, hauser, sunshine @isi.edu