CS259D HW3 Theoretical Topic: Web Security

Since the creation of the first web server in 1990, the World Wide Web has erupted as a medium of information distribution and communication. Today, companies like Google, Facebook, and Amazon employ gigantic web server systems to handle millions to billions of requests per day. However, these web servers have become targets to an exponentially increasing number of cybersecurity attacks. Popular attacks include distributed denial-of-service (DDoS), buffer overflow, and cross site scripting (XSS).

Of course, much technical research has been conducted on how to address these issues and to create prevention and detection mechanisms. As we saw in class, various models can be built around the contents of HTTP requests and web logs. Features such as attribute length, character distribution, access frequency, and program invocation order are analyzed using machine learning (ML) techniques to detect anomalous behavior. In this report, we will attempt to append to this trajectory by surveying research from three more recent papers. The first paper “Server Side Detection of Content Sniffing Attacks” by Barua et al presents an improved algorithm used for detecting malicious uploads to web servers. The second paper “Detection of Insider Attacks to the Web Server” by Choi et al presents a different way of utilizes an algorithm that detects anomalous outsider behavior to instead detect anomalous insider behavior. The third paper “Discriminating DDoS Attacks from Flash Crowds” by Yu et al presents a way of minimizing false positives in existing DDoS detection systems.

The first paper, by Barua et al, discusses content sniffing attacks, a type of XSS attack, in which the browser tries to render non-HTML files that contain malicious HTML or JavaScript, which is generally uploaded by a user to the server. Currently, most browsers employ content sniffing detection algorithms. However, these algorithms are insufficient because they only verify a fixed number of bytes at the beginning of the payload, and they have no way of detecting the malicious impact of the content on browsers. The paper proposes a new detection pipeline with a series of security checks. First, basic checks are done on the MIME type, ensuring the file extension matches the magic header in the payload and ensuring the type conforms to a list of whitelisted MIME types. If this fails, then the file is immediately rejected. Next, the file format is normalized to UTF-8 encoding and is run through an HTML and JavaScript parser. These parsers look for tags in the file signifying HTML or JavaScript code. If such tags exist, the file is flagged as potentially malicious and enters a final check through the mock download tester. The mock download tester emulates a browser and forces the browser to render the file as HTML. If the emulator exhibits any unusual symptoms (i.e. executing JavaScript) while downloading the file, then the file is regarded as malicious and rejected. For its benchmark suite, the paper achieved a false positive rate of 0%, which is the expected rate for current algorithms, and a false negative rate of 0%, which is a marked improvement on current techniques. Overall, the detection system proposed in the paper seems like a clear improvement over current techniques. However, the drawback is that it requires more computation time, which could be prohibitive for larger files or certain applications.

The second paper, by Choi et al,

Paper 2: Detect insider anomalies

Typically, inbound traffic to server is tracked/analyzed for intruder detection

Instead, analyze outbound traffic for anomalies/confidential information

Hybrid detection system – signature-based and anomaly-based detectors

Page 2:

Paper 3: Discriminating DDoS from Flash Crowds

Challenges/Open Problems

False positives are so troublesome

Attacks can come from anywhere, how to keep up with algorithms?

Explored a lot of ML techniques already, how do we improve from here?

Many new techniques are being proposed, but a lot of time computational cost is prohibitive

Conclusion

Overall impression is promising -> work is being done

However, still a great area of need

Undoubtedly, cannot address these problems with current methodologies, so it will be interesting to see what new innovations there will be in the future!

References.

1. “Server Side Detection of Content Sniffing Attacks” by Barua et al, 2011

<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6132950>

1. “Detection of Insider Attacks to the Web Server” by Choi et al, 2012

<http://isyou.info/jowua/papers/jowua-v3n4-3.pdf>

1. “Discriminating DDoS Attacks from Flash Crowds” by Yu et al, 2012

<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6060809>