**Week2 Article Evaluation**  
Nate Bachmeier   
CMGT441: Information Systems Security Management  
University of Phoenix  
May 26, 2014  
Facilitator: Alexander Pons

# Week 2 Article Evaluation

For my article selection, I picked the Microsoft Security Intelligence Report (SIR) Volume 16. The report gives a worldwide perspective on how vulnerabilities are being exploited and general trends around the economics.

## Exploit Trends

The first discussed was once a bug has been discovered the time to remote code execution has significantly declined to under 30 days (Microsoft, 2013). What this means to me is that patch management is critical in today’s world, as soon as an issue has been discovered mitigations need to be put in place.

The next trend was comparing how attackers are getting remote code execution. Back in 2006, stack corruption accounted for 40% of the bugs. However in today’s world smashing the stack has been hunted to extinction, now only accounting for 5%. This is in part due to Address Space Layer Randomization (ASLR) and Data Execution Protection (DEP) being widely deployed requiring “Use after Free Bugs” to become the dominant vector.

To bypass DEP and ASLR the most frequently used methods include using legacy (non-ASLR) binaries and Return Oriented Programming (ROP). Heap Spraying is also widely used which makes sense, given that Use after Free Bugs are being the normal.

Exploit kits are being business to criminal groups with some making over 50,000$ per day. The most common software packages that are being targeted include: Java, Adobe Reader, Internet Explorer, Adobe Flash, and Windows. This makes sense to me given these are widely used programs with a large market share. By focusing on the largest market shares the criminal groups can cast the largest fishing nets.

## World Wide Threat Assessment

In the next section the authors looked at vulnerability counts, pointing out that roughly 1500 medium, 800 high, and 200 low vulnerabilities were discovered in the second half of 2013. What this says to me is that there are a lot of people actively trying to break into our networking systems. They are also finding a lot of problems which might suggest that different adoption of security development lifecycle (SDL) need to be put into place or the SDL needs to be more aggressive.

Of these vulnerabilities most of them are being found in applications that are included outside of the core operating system. The argument that they are making is that other applications are easier find bugs, which means they are not as well engineered.

While I agree that is likely true, this section is not the volume of other applications compared to core OS services. This makes the section suffer from population bias, making it statistically skewed. The same population bias can be seen in their comparison of Microsoft to Other charts.

## Mitigation Risks

The first place to start protecting our systems is with Malware detection software. Microsoft IT uses System Center Endpoint Protection to manage 350,000 devices and is able to keep a 99% compliance rating. Across those same devices there was only about 100 infections during the second half of 2013. This is largely attributed to real time monitoring being deployed to add nodes.

# Conclusions

The SIR provides an interesting and detailed accounting of vulnerabilities and how they are being used. Technologies such as ASLR and DEP have shifted the way that people are able to gain remote code execute forcing a movement toward “Use after Free Bugs.”

To find these bugs people are looking to applications with large market shares, such as Java, Adobe products, and Windows. Currently Java is the single largest source of vulnerabilities being reported each quarter.

In order to mitigate the risks it is important to deploy real time malware monitoring solutions. Having such a system in place can significantly reduce infections and keep the network running healthy and efficiently.

# Bibliography

Microsoft. (2013). *Security Intelligence Report Volumen 16.* Microsoft Corp.