Ryan Cranston, Petal Michaud, Julian Welge

Topic #1 - Attack Vectors of Blockchain

Blockchain has become a large part of our banking and otherwise authentication world. The encryption and methodology behind blockchain has led many people to believe that it is worth fully investing in as it 'can't' be hacked or mistrusted.

Interestingly enough, Blockchain has five attack vectors that could be used to exploit the methodology it is built on. These vectors are Blockchain Network attacks, User Wallets, Smart contact attacks, Transaction verification mechanism attacks, and Mining pool attacks. Our plan is to look at how each of these attack vectors can be exploited and manipulated. We intend on examining what the influence of a successful attack would look like on the Blockchain with the utilization of one or more of these vectors.

We also plan on taking a look at what can be done after an attack on the blockchain and what exactly the recovery process of the blockchain would entail.

Lastly, we plan on discovering ways to prevent these attacks from even being possible as well as any other measures one could implement to keep the blockchain safe and trustworthy. To do this research we will be using a variety of news articles and scholarly articles on the subjects at hand and more specifically examine attacks that have occurred on Coincheck, Verge, and Bancor exchanges.

Topic #2 - Input Validation

A big part of input validation is checking input fields from web forms to ensure that the user's input is valid but it also includes checking data received from other systems through the backend. Improper input validation is a major problem in web development and causes a variety of security issues if it is overlooked. All input should be treated as dangerous in order to prevent security attacks. One single input can be checked for proper length, type, range, format, plausibility, etc.

Web developers often skip input validation because of how tedious and time consuming it can be to write checks for every possible edge case. Languages like JavaScript have built in functions for dealing with input validation making it a top choice for web developers.

For our presentation, we will stress on the importance of input validation and touch on common strategies that are implemented including syntax and semantic validity as well as client side versus server side validation. We will also look at the consequences of improper input validation including SQL injection, buffer overflow and XSS and how to fix them. The presentation will touch on whitelisting and blacklisting data and the issues that arise when doing one over the other. A section of our presentation will focus on ways to validate inputs in Java,

complete with example code and comparisons of what we usually see versus what should be done. We will also mention open source frameworks in Java that help with validation.

Topic #3 - Operating System Security

While there are many ways to keeping our systems safe and secure, such as checking input from the outside world or encrypting sensitive information that we send out involving our systems sensitive information, one of the most effective ways to ensuring data on a system avoids being compromised is by examining security at one of the most fundamental levels, the operating system.

First we intend on looking at some of the many ways OS security can be approached including regular patch updates, installing antivirus engines and software, checking all incoming and outgoing network traffic through firewalls, and creating secure accounts with finite appropriate levels of privilege.

This plane of thinking regarding OS within the security realm is crucial as it practically analyzes security at the highest of authorities for all systems and from this we intend on looking at the different strategies and organization involved to creating a system with as minimal vulnerabilities as possible. We plan on looking at what types of strategies certain companies like IBM, Microsoft, and Apple have done to keep their systems safe and trustworthy as well as potentially some more innovative routes we hope to come across. For example just glancing at IBM's page they include sections discussing operating system minimization, logging and monitoring, and denying access by default.

If we find that most of the material between these strategies are common across the board, we also plan to look into how one could potentially recover a compromised system and safe guards that we should be implementing should the system be compromised.