Howard Poston

Blockchain & Cyber Security

Mapping the OWASP Top Ten to Blockchain

February 12, 2019

In my Blockchain Security course, I (unsurprisingly) get a lot of students with a background in cyber security. As a result, I have been asked several times how well the Open Web Application Security Project's (OWASP) Top Ten list for web application vulnerabilities maps to the blockchain space. In this article, we'll explore the OWASP Top Ten list and, where possible, point out areas where blockchain technology is potentially or actually vulnerable.

Injection

In an injection attack, an attacker takes advantage of poor sanitization of user input to attack a system. If the software developer did not appropriately handle the user input, it can be crafted in a way that allows it to run unauthorized commands. For example, failing to sanitize SQL input could allow a user to bypass authentication and run unauthorized queries against the database.

Poor input sanitization is definitely a potential issue in blockchain technology. Prior to the launch of the EOS mainnet, researchers at Qihoo 360 reported a vulnerability in the EOS smart contract parsing function that allowed a buffer-out-of-bounds write1. The researchers developed a

proof of concept where the vulnerability allowed them to break out of the EOS sandbox and launch a reverse shell on the infected machine. If this vulnerability had been exploited on a running network, it could have allowed the attacker to compromise every node on the network as they ran the malicious smart contract once it was included in a valid block.

Broken Authentication

Broken authentication is a general vulnerability referring to any issues in the implementation of an authentication mechanism in an application. These issues could allow an attacker to masquerade as a legitimate user on a temporary or permanent basis.

Proper implementation of authentication functionality is vital to the proper function of the blockchain system, and the wide use of public key cryptography means that a large attack surface exists.

The LISK cryptocurrency is a good example of where design oversights allowed an attack on authentication in a blockchain. In LISK, a user's address on the blockchain is achieved by hashing their public key and truncating the result to 64 bits2. This hash and truncation method of generating addresses is common; however, the short length of LISK addresses and the fact that

Howard Poston

Blockchain & Cyber Security

addresses are not immediately tied to public keys makes LISK vulnerable to attack.

Addresses in LISK are only bound to public keys when a user initiates a transaction by either sending value out of the account or voting for a delegate. As a result, many accounts that had only received value were vulnerable to attack. An attacker only has to perform 264 key generation operations to find a private/public keypair that would map to a given address. Targeting any of M addresses decreases the complexity of finding any match by a factor of M. This attack is definitely feasible, and some vulnerable accounts held millions of dollars.

Sensitive Data Exposure

Sensitive data exposure is a self-explanatory vulnerability. If an application holds valuable data that must be kept secret, this data needs to be appropriately protected.

Blockchain technology is largely vulnerable to this vulnerability due to a lack of understanding around the technology. The blockchain is immutable, meaning that any data stored on it cannot be removed (without control of every node in the network). Most blockchains are also public, allowing anyone to download and store a complete copy of the data in the ledger.

In the short term, this combination makes blockchains vulnerable to data mining efforts. Many organizations specialize in mining the blockchain for public data that can aggregated into useful information. This data can be used for law enforcement, corporate espionage, and other purposes.

In the long term, all cryptography is breakable. Quantum computing is on the horizon and, while it won't break blockchain technology, it will allow the decryption of any data stored on the blockchain that's encrypted with public key cryptography. If data needs to be kept private forever (like personal data protected by privacy laws), don't put it on a public blockchain.

XML External Entities (XXE)

XML External Entities (XXE) are vulnerabilities based upon external entity references inside XML documents. The risk of these is that sensitive internal files stored on the webserver may be accessible using these references. Since blockchain is not XML-based, this vulnerability does not really apply.

Broken Access Control

Broken access control is similar to but distinct from broken authentication. In broken authentication, an attacker pretends to be an unauthorized user, while, in broken access control, a

Howard Poston

Blockchain & Cyber Security

malicious user gains unauthorized access to protected functionality.

Poorly implemented access control mechanisms are one of the major vulnerabilities seen in Ethereum smart contracts. The Parity smart contract-based multi-signature wallet is known for being exploited twice due to access control vulnerabilities. In both cases, Parity smart contracts had a function designed to let the owner call it and claim ownership of the contract but didn't check that it was only called once.

In the first attack, an attacker called this function to take control of Parity wallet contracts and drain them of the stored funds. In the second, a similar function in a library contract used by all Parity wallets was attacked. The attacker took control and then self-destructed the function, making all Parity wallets unusable and causing about 1% of all Ether to be lost forever3.

Security Misconfiguration

Security misconfigurations is another general OWASP vulnerability. It refers to using insecure default configurations of software or using configurations that make the system vulnerable to attack. As such, it is one of the most commonly seen types of vulnerabilities.

Blockchains are implemented as software running on client's machines in a peer-to-peer network, so it makes

sense that security misconfigurations could impact security. In one case, users of an Ethereum wallet configured their wallets to listen and accept external commands via RPC (port 8545). Attackers taking advantage of this vulnerability were able to steal $20 million worth of Ether4.

Cross-Site Scripting (XSS)

A website is vulnerable to a cross-site scripting attack if it includes untrusted data within a webpage without validating and sanitizing it first. This vulnerability allows an attacker to run scripts within a victim's browser.

Cross-site scripting vulnerabilities can affect blockchain systems in a couple of different ways. Cross-site scripting vulnerabilities have been exploited in other software to allow cryptomining malware to be run on the victim's computer.

Cross-site scripting vulnerabilities can affect blockchain security more directly if they exist in blockchain explorers. Blockchain explorers have display transaction data, which is untrusted data potentially under an attacker's control. If an XSS vulnerability exists in a combined blockchain explorer and wallet, exploitation of the explorer could allow access to a user's private key and control over their account5.

Insecure Deserialization

Serialization is commonly used for transmission of sets of data across a network. If deserialization code is improperly implemented, a malicious transmission could allow an attacker to exploit a vulnerable machine.

While no reported attacks have taken advantage of deserialization vulnerabilities, blockchain systems commonly use serialization for transmission of transactions. Since transaction data is under the control of (potentially malicious) users, vulnerable deserialization code could lead to compromise of blockchain systems.

Using Components with Known Vulnerabilities

Most software is built on top of or reuses other software. If these dependencies contain vulnerabilities, the software using them may be exploited by attackers targeting these vulnerabilities. The Equifax hack is a great example of how vulnerable third-party code can have significant impacts on an organization's security.

Code reuse in Ethereum smart contracts is even more common than non-blockchain applications. In fact, less than 10% of Ethereum smart contracts do not reuse code6. Since many smart contract programmers have limited experience with the technology and the associated risks, this means that many smart contracts on the Ethereum blockchain contain

known vulnerabilities due to code reuse.

Insufficient Logging and Monitoring

While secure development processes are important, they're only half the battle. Once a system has been deployed, it is also important to log and monitor events on the system for abnormalities that may signal an attack. Failure to do so may leave the system vulnerable to exploitation via attack vectors overlooked during the design process.

The blockchain creates an immutable ledger of actions taken on the system, making it ideal for logging purposes. However, while logging is great, it's useless is no-one is looking at the logs. Many smart contracts on the blockchain are "fire and forget" and go unmonitored by their owners, making them potentially vulnerable to exploitation without detection.

Securing the Blockchain

The OWASP Top Ten is designed to inform developers of the most common security mistakes made in web development. While blockchain systems are not traditional web applications, many of the same vulnerabilities apply. Of the vulnerabilities listed in the Top Ten list, only XXE is not directly applicable to some component of the blockchain ecosystem.

Howard Poston

Blockchain & Cyber Security

While the OWASP Top Ten is a good starting point when developing blockchain systems and smart contracts, the blockchain ecosystem creates additional potential security issues. The Decentralized Application Security Project (DASP) maintains a similar Top Ten list geared toward educating smart contract developers about the most common mistakes made on the Ethereum platform. Understanding how the blockchain ecosystem works and the security assumptions made at each level are also a vital part of ensuring holistic distributed ledger security.

Sources

[1] https://thehackernews.com/2018/05/eos-blockchain-smart-contract.html

[2] https://research.kudelskisecurity.com/2018/01/16/blockchains-how-to-steal-millions-in-264-operations/

[3] https://www.theregister.co.uk/2017/11/10/parity_280m_ethereum_wallet_lockdown_hack/

[4] https://cointelegraph.com/news/report-misconfigured-ethereum-clients-have-resulted-in-hack-of-around-20-mln

[5] https://blockpath.com/r/Interesting/comments/1g/the_transaction_that_can_xss_attack_unprepared/

[6]

http://www.ccs.neu.edu/home/amislove/publications/Ethereum-IMC.pdf

Prev / Next

Howard Poston

Blockchain & Cyber Security

About

Experience

Contact

Blog