Summary

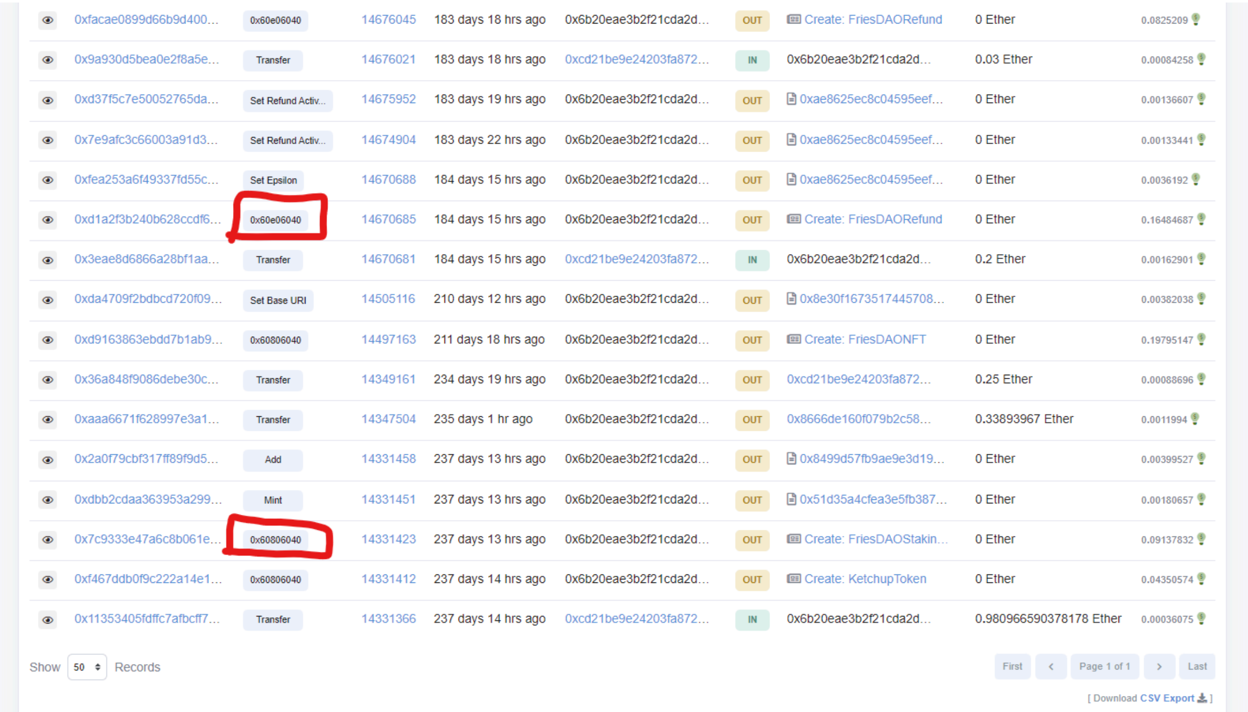
On October 27th, 5:58PM UTC, friesDAO contracts were exploited by an attacker taking control of our own deployer address through a profanity attack vector. The hacker was able to drain the treasury of its USDC through the refund contract, drain the FRIES tokens in the staking contract, subsequently selling it all into the Uniswap pool. All transactions in the main attack with the refund contract were confirmed in the same block, then three hours later, the attacker came back for the staking pool

1. Vulnerability

Some friesDAO contracts including KCHUP, StakingPool, NFT, and Refund, were deployed by one address over the course of development: 0x6B20EAE3B2F21cDA2d5a8EA123AE262C86a6DF99

This address was generated for KCHUP (0x51D35a4cfea3e5fb387e467d31cc0c87f6038a) to have a vanity address (51D35 = “SIDES”) using Profanity, a local multithreaded GPU vanity address miner that was considered safe at the time of generation. Profanity has options to generate a deployer address such that the first contract it deploys will have the address desired.

However, ownership of the contracts had not been transferred to a different address such as the multisig after deployment in case of any changes or bugs needed, specifically due to the high risk of how the refund contract interacts with funds. Thus it was determined that it was safer to leave room for emergency changes and that considering our primary developer Slip was internally doxxed, that any attempt of theft would immediately implicate the developer. In fact the initial deployment of the refund contract had issues and had been redeployed to fix a calculation error:



0xfacae0899d66b9d400...	0x60e06040	14676045	183 days 18 hrs ago	0x6b20eae3b2f21cda2d...	OUT	Create: FriesDAORefund	0 Ether	0.0825209
0x9a930d5bea0e2f8a5e...	Transfer	14676021	183 days 18 hrs ago	0xcd21be9e24203fa872...	IN	0x6b20eae3b2f21cda2d...	0.03 Ether	0.00084258
0xd37f5c7e50052765da...	Set Refund Activ...	14675952	183 days 19 hrs ago	0x6b20eae3b2f21cda2d...	OUT	0xae8625ec8c04595eef...	0 Ether	0.00136607
0x7e9afc3c66003a91d3...	Set Refund Activ...	14674904	183 days 22 hrs ago	0x6b20eae3b2f21cda2d...	OUT	0xae8625ec8c04595eef...	0 Ether	0.00133441
0xfea253a6f49337fd55c...	Set Epsilon	14670688	184 days 15 hrs ago	0x6b20eae3b2f21cda2d...	OUT	0xae8625ec8c04595eef...	0 Ether	0.0036192
0xd1a2f3b240b628ccdf6...	0x60e06040	14670685	184 days 15 hrs ago	0x6b20eae3b2f21cda2d...	OUT	Create: FriesDAORefund	0 Ether	0.16484687
0x3eae8d6866a28bf1aa...	Transfer	14670681	184 days 15 hrs ago	0xcd21be9e24203fa872...	IN	0x6b20eae3b2f21cda2d...	0.2 Ether	0.00162901
0xda4709f2bdbcd720f09...	Set Base URI	14505116	210 days 12 hrs ago	0x6b20eae3b2f21cda2d...	OUT	0x8e30f1673517445708...	0 Ether	0.00382038
0xd9163863ebdd7b1ab9...	0x60806040	14497163	211 days 18 hrs ago	0x6b20eae3b2f21cda2d...	OUT	Create: FriesDAONFT	0 Ether	0.19795147
0x36a848f9086debe30c...	Transfer	14349161	234 days 19 hrs ago	0x6b20eae3b2f21cda2d...	OUT	0xcd21be9e24203fa872...	0.25 Ether	0.00088696
0xaaa6671f628997e3a1...	Transfer	14347504	235 days 1 hr ago	0x6b20eae3b2f21cda2d...	OUT	0x8666de160f079b2c58...	0.33893967 Ether	0.0011994
0x2a0f79cbf317ff89f9d5...	Add	14331458	237 days 13 hrs ago	0x6b20eae3b2f21cda2d...	OUT	0x8499d57fb9ae9e3d19...	0 Ether	0.00399527
0xdbb2cdaa363953a299...	Mint	14331451	237 days 13 hrs ago	0x6b20eae3b2f21cda2d...	OUT	0x51d35a4cfea3e5fb387...	0 Ether	0.00180657
0x7c9333e47a6c8b061e...	0x60806040	14331423	237 days 13 hrs ago	0x6b20eae3b2f21cda2d...	OUT	Create: FriesDAOStakin...	0 Ether	0.09137832
0xf467ddb0f9c222a14e1...	0x60806040	14331412	237 days 14 hrs ago	0x6b20eae3b2f21cda2d...	OUT	Create: KetchupToken	0 Ether	0.04350574
0x11353405fdffc7afbcff7...	Transfer	14331366	237 days 14 hrs ago	0xcd21be9e24203fa872...	IN	0x6b20eae3b2f21cda2d...	0.980966590378178 Ether	0.00036075

As time progressed and the contracts appeared to be working properly, the developer unfortunately forgot to transfer ownership of these contracts to the multisig and had assumed they were already transferred when in reality, the deployer address (0x6B20) still had full ownership and control over these contracts. Note that the deployer address' private key never left the metamask and was never exported out in any external format including to the developer himself.

It is possible that the way the attacker got the private key was first by guessing that the deployer address was a vanity address through implication of the vanity "SIDES" contract address for KCHUP.

Subsequently, the attacker brute-forced the private key using profanity's now known vulnerabilities, which dramatically reduces the possibilities of private keys due to flaws in generation and is susceptible to even consumer grade computing power. Learn more about the Profanity hack at: https://medium.com/amber-group/exploiting-the-profanity-flaw-e986576de7ab

What is interesting to note, however, is that 0x51D35 ("sides") is unlikely to appear immediately obvious and would have been hard to guess for a random hacker. Furthermore, this was never mentioned publicly in any channels. Additionally there were some interactions of this hacker's wallet with other known wallets, who happen to also be a DAO member. These contributed to some of our investigation angles.

2. Attack

The attack was two-part, complex and required a deep understanding of our contracts.

After gaining control of the deployer contract, the first part drained all of the USDC from the treasury for 2,138,705.403949 USDC:

1. The attacker sent in some eth for gas to the deployer contract and the exploit
2. Swapped it to a bit of FRIES tokens
3. Set the manual, fixed refund rate variable to a high number used for locking the automated refund rate when we needed to move funds in the multisig
4. Set the manual refund rate active
5. Changed the merkle root whitelist of the NFT to include his deployer address (very difficult/annoying and requires reading the format of the whitelist carefully)
6. Minted a founder's edition NFT to be enable refund capability
7. Refunded the small bit of purchased FRIES token for the entire treasury's USDC
8. Transferred USDC to his wallet
9. Transferred remaining ETH gas to his wallet

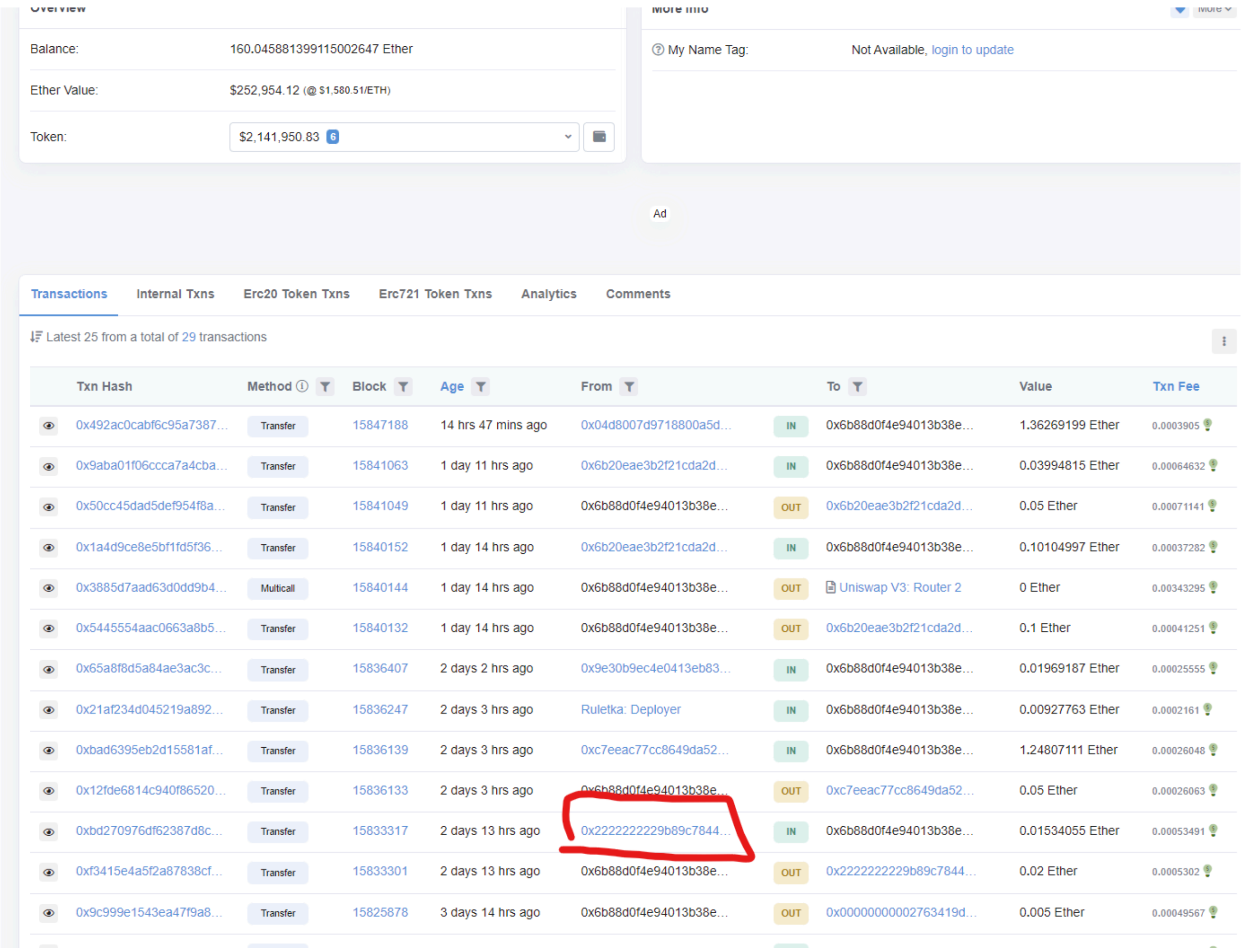
The second part took all of the FRIES out of the staking pool, then sold them through Uniswap to extract USDC from the liquidity pool for 120.128930112550592565 ETH ($189,954.761991 at the time):

1. The attacker again, sent in some eth for gas
2. Took all of the FRIES tokens out of the staking pool using "governanceRecoverUnsupported" which is a standard MasterChef method supposed to be used to recover ERC20 tokens that are mistakenly sent to the pool.
3. Swapped these FRIES tokens to ETH with a direct send to the attacker wallet
4. Transferred remaining ETH gas to his wallet

The attacker again also drained the new 0.01 ETH gas we sent in afterwardnwhich was used to transfer ownership of all of our contracts back to the multisig after the exploit.

3. Attacker(s)

The attacker (0x6b88d0f4e94013b38e7c49ddc24135bfb0e2d49b) had already been exploiting projects and users using the same profanity method before our attack. One of the wallet interactions is with 0x2222222229b89c7844f19ef503c4dc503be47f84, which is associated with a known user and also a member of friesDAO with a history of questionable coding including sandwich botting and black hat activities. Furthermore, this user has been implicated in questionable actions in past projects and has had a history of brazenly challenging others' allegations. However this user's wallet appears to also be drained of its dust, which is possible that it is due to also a Profanity wallet generation and was exploited. Although the evidence is circumstantial, it is odd that the interaction/exploit of 0x22222 occurred just one day prior to the hack while also being a member of friesDAO and so this user remains a person of interest.



Overview

Balance:	160.045881399115002647 Ether
Ether Value:	$252,954.12 (@ $1,580.51/ETH)
Token:	$2,141,950.83 6

More Info

My Name Tag:	Not Available, login to update

Ad

Latest 25 from a total of 29 transactions

Txn Hash	Method	Block	Age	From		To	Value	Txn Fee
0x492ac0cabf6c95a7387...	Transfer	15847188	14 hrs 47 mins ago	0x04d8007d9718800a5d...	IN	0x6b88d0f4e94013b38e...	1.36269199 Ether	0.0003905
0x9aba01f06ccca7a4cba...	Transfer	15841063	1 day 11 hrs ago	0x6b20eae3b2f21cda2d...	IN	0x6b88d0f4e94013b38e...	0.03994815 Ether	0.00064632
0x50cc45dad5def954f8a...	Transfer	15841049	1 day 11 hrs ago	0x6b88d0f4e94013b38e...	OUT	0x6b20eae3b2f21cda2d...	0.05 Ether	0.00071141
0x1a4d9ce8e5bf1fd5f36...	Transfer	15840152	1 day 14 hrs ago	0x6b20eae3b2f21cda2d...	IN	0x6b88d0f4e94013b38e...	0.10104997 Ether	0.00037282
0x3885d7aad63d0dd9b4...	Multicall	15840144	1 day 14 hrs ago	0x6b88d0f4e94013b38e...	OUT	Uniswap V3: Router 2	0 Ether	0.00343295
0x5445554aac0663a8b5...	Transfer	15840132	1 day 14 hrs ago	0x6b88d0f4e94013b38e...	OUT	0x6b20eae3b2f21cda2d...	0.1 Ether	0.00041251
0x65a8f8d5a84ae3ac3c...	Transfer	15836407	2 days 2 hrs ago	0x9e30b9ec4e0413eb83...	IN	0x6b88d0f4e94013b38e...	0.01969187 Ether	0.00025555
0x21af234d045219a892...	Transfer	15836247	2 days 3 hrs ago	Ruletka: Deployer	IN	0x6b88d0f4e94013b38e...	0.00927763 Ether	0.0002161
0xbad6395eb2d15581af...	Transfer	15836139	2 days 3 hrs ago	0xc7eeac77cc8649da52...	IN	0x6b88d0f4e94013b38e...	1.24807111 Ether	0.00026048
0x12fde6814c940f86520...	Transfer	15836133	2 days 3 hrs ago	0x6b88d0f4e94013b38e...	OUT	0xc7eeac77cc8649da52...	0.05 Ether	0.00026063
0xbd270976df62387d8c...	Transfer	15833317	2 days 13 hrs ago	0x2222222229b89c7844...	IN	0x6b88d0f4e94013b38e...	0.01534055 Ether	0.00053491
0xf3415e4a5f2a87838cf...	Transfer	15833301	2 days 13 hrs ago	0x6b88d0f4e94013b38e...	OUT	0x2222222229b89c7844...	0.02 Ether	0.0005302
0x9c999e1543ea47f9a8...	Transfer	15825878	3 days 14 hrs ago	0x6b88d0f4e94013b38e...	OUT	0x00000000002763419d...	0.005 Ether	0.00049567

However, we recognize that it is entirely possible a third party entity studied the tokens of the 0x2222 user, and noticed the vanity address of the FRIES token itself (0xFA57F00D: "FastFood") and while this token is protected by multisig ownership it may have given enough reason for the user to sniff out other related contracts of the DAO, begin brute forcing attempts, and study the contract code. It remains a mystery that if this was indeed a targeted attempt by an outsider, it would take time to go through 0x2222's tokens, brute force test the contracts successfully to ensure worthwhile time, then study all the protocol mechanic information in the contract code, all within 1 day. If this preparation took even longer (and the exploit of 0x2222 is a byproduct of targeting friesDAO first), then why bother draining other dust accounts in the interim that are not nearly at the same level of sophistication?

Thus we suspect the likelihood is higher that someone who has been around friesDAO for longer had planned and prepared for this attack. We do also recognize the suspicion of the primary developer having the potential to conduct this attack, but likelihood is small due to voluntarily self doxxing to other members of the team, knows we are close to closing a store deal to get considerable amounts of vested tokens, and would have otherwise had the ability to carry out an attack a month earlier when the Profanity hack was revealed due to self knowledge that Profanity was used.

4. Next Steps

This is still an ongoing investigation and we invite members and the public to help investigate the on chain analysis as well. Because we are a US entity we have the obligation to file a report with the FBI's IC3/cyber crimes unit for further assistance. Of course, we do also invite the hacker, if reading this, to anonymously return the funds to the multisig to mitigate our law enforcement efforts. We are also open to dialogue should you wish to reply to the friesDAO twitter account (however any funds should be returned directly to the multisig, anywhere else may be a scam).

We recognize the inattentive series of errors that lead to this event, of not using extra diligence in revisiting the Profanity generated contracts when this exploit became public knowledge. Going forward we have implemented a plan for a secondary developer to always check all contract code and deployments no matter how simple the process may seem in case of any oversight that was not communicated to other team members. We have set alerts to currently watch the hacker's wallet for any movement of our funds, especially to a CEX which we can doxx, and we also encourage others to do so as well. (wallet containing our stolen funds is https://etherscan.io/address/0x6b88d0f4e94013b38e7c49ddc24135bfb0e2d49b). You can also report to Etherscan to flag this wallet as an exploiter.

If funds cannot be retrieved successfully soon, there may be a possibility to also raise a small amount to secure the store (this is still subject to the landlord approval of the current deal we are working on) so that we may continue our endeavors and drive recovery through the FRIES token and our FRIES treasury. The Uniswap liquidity pool will also be restored accordingly once we determine the proper tokenomics.