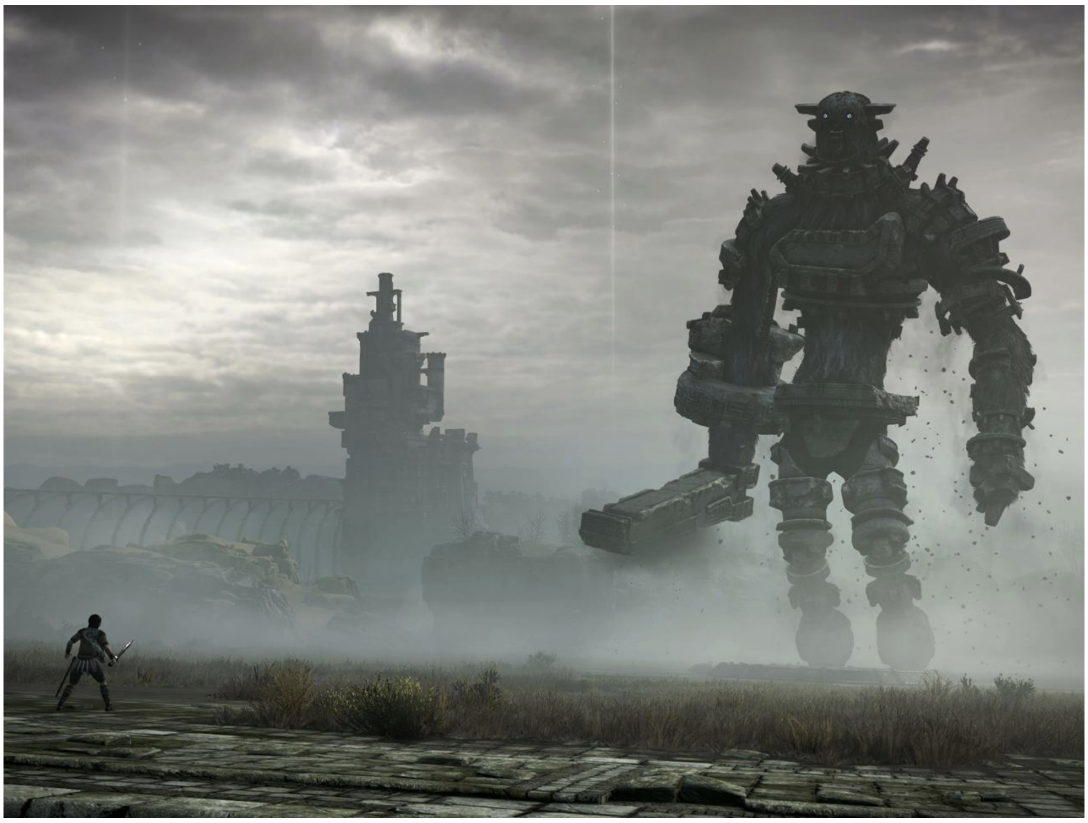
eth2 Red Team - Report o

methodology and scope



eth2 red teaming in an image

breadth-first a.k.a. trolling all over the place

Eth2 has been under active research and development for several years before being successfully launched this year. It is likely one of the largest open-source projects of all time, driven and developed by a diverse, decentralized community. As a result, standards and best practices for red teaming and pen testing efforts on behalf of centralized organizations have only limited validity in the context of eth2.

As both relevant information and the implementation of the eth2 system is distributed across a variety of resources and as the bootstrapping of the red team research overlapped with the launch of eth2, we chose a breadth-first methodology to develop cross-domain and knowledge and identify potential weakest links in eth2's security supply chain to inform future research.

The chosen methodology also generates ”real world” data on how resource and time-constrained attackers perceive and interact maliciously with the evolving eth2 ecosystem.

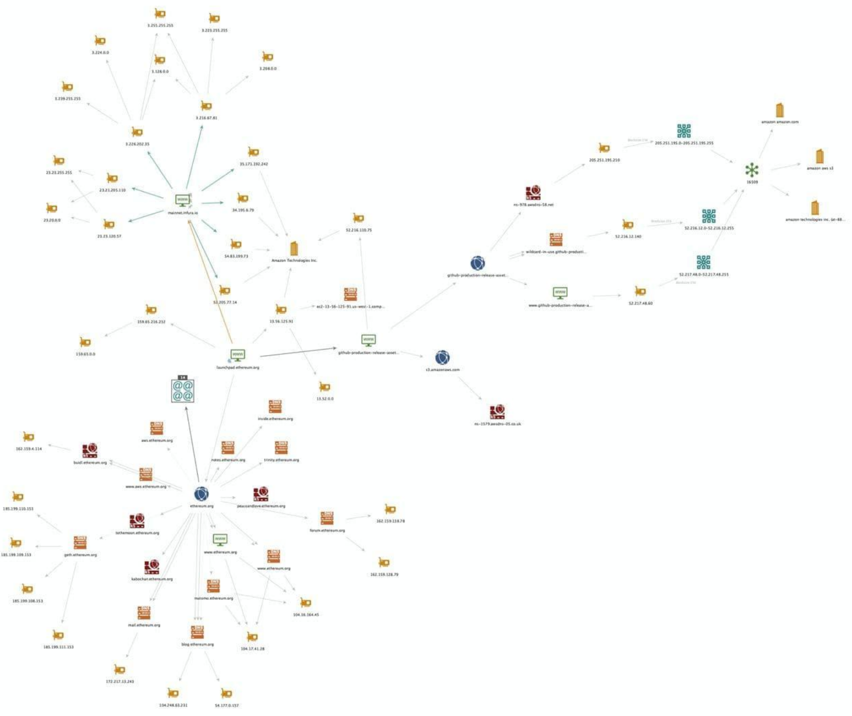
findings and non-findings

launchpad.ethereum.org

Scope

The eth2 launchpad was briefly researched and tested during the launch, as it is a high-value target for various attacks. Due to its reliance on third-party services (particularly services offered by Infura, Github, and AWS), the auxiliary services were included in the investigation, but had to be outscoped for legal reasons in the course of the analysis.

As a first step, we performed an OSINT analysis of the launchpad to understand and prepare target selection and future investigations.



From this set, we selected essentially 3 targets for a brief analysis

1. The AWS s3 bucket used by Github to host production releases of the eth2.0-deposit-CLI tool at github-production-release-asset-2e65be.s3.amazonaws.com
2. The Infura endpoint used to fetch deposits for the launchpad at mainnet.infura.io/v3/9a011a6cf597453481cc94c05ba7dc54
3. The Netlify instance hosting the launchpad itself at silly-engelbart-23669a.netlify.app

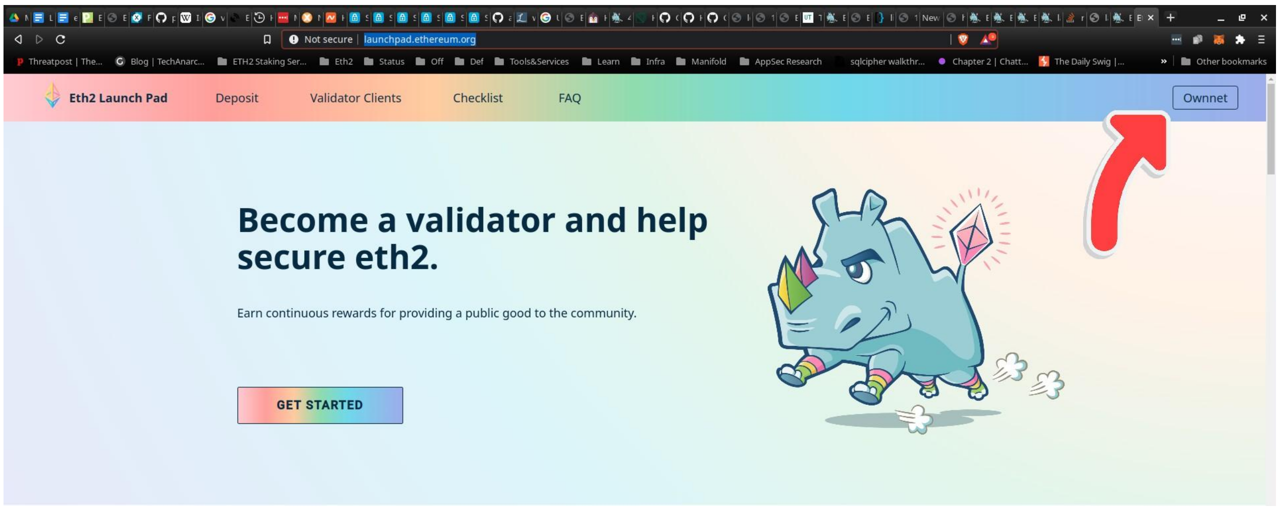
Results

Local Testing

Initially, we tested the launchpad and its auxiliary services in a setting where an attacker had access to the WLAN of the victim machine. In this setup, the launchpad is not safe to use and a compromise to a large extent was possible (delivering an arbitrary website with an altered deposit address or a malicious eth2-deposit-cli). Exploitable vulnerabilities are mainly but not exclusively caused by the auxiliary endpoints being mis/under configured.

	ssl strip	dns spoofing	SSL security summary
AWS S3	partially vulnerable	vulnerable	https://www.ssllabs.com/ssltest/analyze.html?d=github-production-release-asset-2e65be.s3.amazonaws.com
Infura	partially vulnerable	vulnerable	https://www.ssllabs.com/ssltest/analyze.html?d=mainnet.infura.io
Launchpad	not vulnerable	vulnerable	https://www.ssllabs.com/ssltest/analyze.html?d=silly-engelbart-23669a.netlify.app

Only the netlify instance was configured securely and included HSTS to prevent MITM SSL stripping attacks.
However, a combination of arp+DNS spoofing allows an attacker to redirect a victim machine (tested on Ubuntu 20.4, Brave, Firefox, Chrome) to an arbitrary site bypassing the HSTS.


Not secure | launchpad.ethereum.org
Eth2 Launch Pad
Deposit
Validator Clients
Checklist
FAQ
Ownnet
Become a validator and help secure eth2.
Earn continuous rewards for providing a public good to the community.
GET STARTED

Remote testing

We conducted brief infrastructure tests to understand potential, potentially scalable remote threats by a less privileged attacker. At some point, we discontinued the research due to legal risks. Up until then, no findings were made, i.e., the launchpad's security in this threat model is summarized in the graph below :)

The following table documents testing efforts and summarizes the non-findings:

	Manual audit	hidden site/directory fuzzing	Port probing	http-request smuggling	Server cache poisoning	Metasploit, etc.
AWS S3	Not applicable	No signal	No signal	Limited signal	discontinued	untested
mainnet.infura	Not applicable	No signal	untested	No signal	untested	untested
Launchpad	1 false positive	No signal	No signal	No signal	untested	untested

cross client issue - libp2p is susceptible to TCP-reset attacks

Location
https://github.com/prysmaticlabs/prysm/blob/develop/beacon-chain/p2p/handshake.go#L153

Synopsis
An attacker can learn the IP addresses and open TCP ports via libp2p. A malicious actor may send large volumes of TCP packets with spoofed source/destination IP and TCP "RST" set to true. Source and destination nodes will disconnect their TCP streams when receiving and processing the spoofed packets **iff** the packet's sequence numbers are in the correct TCP window size.

Therefore, there are three subtypes for this attack to consider;

a) Blind TCP-reset attacks;

   This subtype is essentially >= TCP based DOS, where an attacker spams the libp2p network with TCP reset packets with arbitrary sequence numbers.

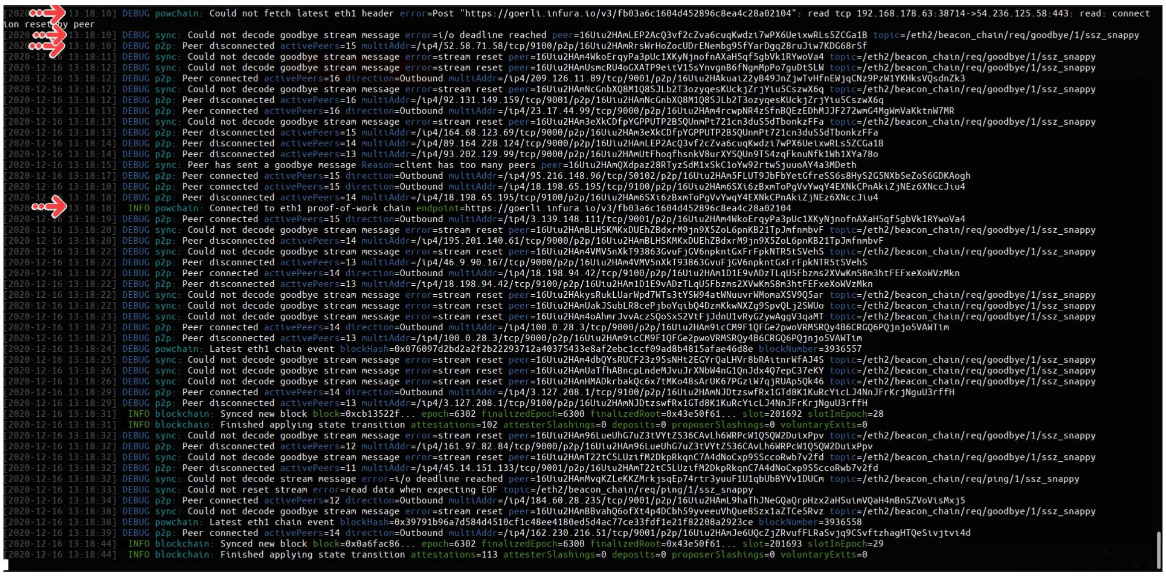
b) Educated TCP-reset attacks; an attacker with a data science hat. Interesting entry points for this attack are:
   - a newly restarted node (rather low entropy for the attack as low sequence numbers, small amount of connections)
   - Established and fully synced nodes with long-lived connections (TCP packets/sequence numbers per connected hour/day/month are observable by connecting to victim nodes through libp2p)

c) Responsive TCP-reset attacks;

   A malicious actor can perform this attack, if it can directly observe the TCP stream, i.e. by an actor with access to LAN, VPS providers, ISPs

Impact

The naive PoC of a responsive TCP-reset attack had significant impact on all connections of the Prysm beacon node:



```
[2020      13:18:10] DEBUG powchain: Could not fetch latest eth1 header error=Post "https://goerli.infura.io/v3/fb03a6c1604d452896c8ea4c28a02104": read tcp 192.168.178.63:38714->54.236.125.58:443: read: connect
ion reset by peer
[2020      13:18:10] DEBUG sync: Could not decode goodbye stream message error=i/o deadline reached peer=16Uiu2HAmLEP2AcQ3vf2cZva6cuqKwdzi7wPX6UeixwRLs5ZCGa1B topic=/eth2/beacon_chain/req/goodbye/1/ssz_snappy
[2020      13:18:10] DEBUG p2p: Peer disconnected activePeers=15 multiAddr=/ip4/52.58.71.58/tcp/9100/p2p/16Uiu2HAmRrsWrHoZocUDrENembg95fYarDgq28ruJiw7KDG68rSf
[2020-12-16 13:18:11] DEBUG sync: Could not decode goodbye stream message error=stream reset peer=16Uiu2HAm4WkoErqyPa3pUc1XKyNjnofnAXaH5qf5gbVk1RYwoVa4 topic=/eth2/beacon_chain/req/goodbye/1/ssz_snappy
[2020-12-16 13:18:12] DEBUG sync: Could not decode goodbye stream message error=stream reset peer=16Uiu2HAmUsmcRU4oGXATP9eitV15sYnvgnB6fNgmMpPo7guDtSLW topic=/eth2/beacon_chain/req/goodbye/1/ssz_snappy
[2020-12-16 13:18:12] DEBUG p2p: Peer connected activePeers=16 direction=Outbound multiAddr=/ip4/209.126.11.89/tcp/9001/p2p/16Uiu2HAkuai22yB49JnZjwTvHfnEWjqCNz9PzW1YKHksVQsdnZk3
[2020-12-16 13:18:12] DEBUG sync: Could not decode goodbye stream message error=stream reset peer=16Uiu2HAmNcGnbXQ8M1Q8SJLb2T3ozyqesKUckjZrjYiu5CszwX6q topic=/eth2/beacon_chain/req/goodbye/1/ssz_snappy
[2020-12-16 13:18:12] DEBUG p2p: Peer disconnected activePeers=16 multiAddr=/ip4/92.131.149.159/tcp/9001/p2p/16Uiu2HAmNcGnbXQ8M1Q8SJLb2T3ozyqesKUckjZrjYiu5CszwX6q
[2020-12-16 13:18:13] DEBUG p2p: Peer connected activePeers=16 direction=Outbound multiAddr=/ip4/23.17.44.99/tcp/9000/p2p/16Uiu2HAm4rcwpNR4zSfnBQEzEDhMJJF272wmG4MgWmVaKktnW7MR
[2020-12-16 13:18:13] DEBUG sync: Could not decode goodbye stream message error=stream reset peer=16Uiu2HAm3eXkCDfpYGPPUTP2B5QUnmPt721cn3duS5dTbonkzFFa topic=/eth2/beacon_chain/req/goodbye/1/ssz_snappy
[2020-12-16 13:18:13] DEBUG p2p: Peer disconnected activePeers=15 multiAddr=/ip4/164.68.123.69/tcp/9000/p2p/16Uiu2HAm3eXkCDfpYGPPUTP2B5QUnmPt721cn3duS5dTbonkzFFa
[2020-12-16 13:18:14] DEBUG p2p: Peer disconnected activePeers=14 multiAddr=/ip4/89.164.228.124/tcp/9000/p2p/16Uiu2HAmLEP2AcQ3vf2cZva6cuqKwdzi7wPX6UeixwRLs5ZCGa1B
[2020-12-16 13:18:14] DEBUG p2p: Peer disconnected activePeers=13 multiAddr=/ip4/93.202.129.99/tcp/9000/p2p/16Uiu2HAmUtFhoqfhsnkV8urXYSQUn9TS4zqFknuNfk1Wh1XYa78o
[2020-12-16 13:18:15] DEBUG sync: Peer has sent a goodbye message Reason=client has too many peers peer=16Uiu2HAmQXdpaz28RTyzSdM1xSkC1oYw92rtw5juuoAY4a3MDeth
[2020-12-16 13:18:17] DEBUG p2p: Peer connected activePeers=15 direction=Outbound multiAddr=/ip4/95.216.148.96/tcp/50102/p2p/16Uiu2HAm5FLUT9JbFbYetGfreSS6s8HyS2G5NXbSeZoS6GDKAogh
[2020-12-16 13:18:18] DEBUG p2p: Peer connected activePeers=15 direction=Outbound multiAddr=/ip4/18.198.65.195/tcp/9100/p2p/16Uiu2HAm6SXi6zBxmToPgVvYwqY4EXNkCPnAkiZjNEz6XNccJiu4
[2020-12-  6 13:18:18] DEBUG p2p: Peer disconnected activePeers=14 multiAddr=/ip4/18.198.65.195/tcp/9100/p2p/16Uiu2HAm6SXi6zBxmToPgVvYwqY4EXNkCPnAkiZjNEz6XNccJiu4
[2020      13:18:18]  INFO powchain: Connected to eth1 proof-of-work chain endpoint=https://goerli.infura.io/v3/fb03a6c1604d452896c8ea4c28a02104
[2020-12-  6 13:18:19] DEBUG p2p: Peer connected activePeers=15 direction=Outbound multiAddr=/ip4/3.139.148.111/tcp/9001/p2p/16Uiu2HAm4WkoErqyPa3pUc1XKyNjnofnAXaH5qf5gbVk1RYwoVa4
[2020-12-16 13:18:20] DEBUG sync: Could not decode goodbye stream message error=stream reset peer=16Uiu2HAmBLHSKMKxDUEhZBdxrM9jn9X5ZoL6pnKB21TpJmfnmbvF topic=/eth2/beacon_chain/req/goodbye/1/ssz_snappy
[2020-12-16 13:18:20] DEBUG p2p: Peer disconnected activePeers=14 multiAddr=/ip4/195.201.140.61/tcp/9000/p2p/16Uiu2HAmBLHSKMKxDUEhZBdxrM9jn9X5ZoL6pnKB21TpJmfnmbvF
[2020-12-16 13:18:22] DEBUG sync: Could not decode goodbye stream message error=stream reset peer=16Uiu2HAm4VMV5nXkT93863GvuFjGV6npkntGxFrFpkNTR5tSVehS topic=/eth2/beacon_chain/req/goodbye/1/ssz_snappy
[2020-12-16 13:18:22] DEBUG p2p: Peer disconnected activePeers=13 multiAddr=/ip4/46.9.90.167/tcp/9000/p2p/16Uiu2HAm4VMV5nXkT93863GvuFjGV6npkntGxFrFpkNTR5tSVehS
[2020-12-16 13:18:22] DEBUG p2p: Peer connected activePeers=14 direction=Outbound multiAddr=/ip4/18.198.94.42/tcp/9100/p2p/16Uiu2HAm1D1E9vADzTLqU5Fbzms2XVwKmS8m3htFEFxeXoWVzMkn
[2020-12-16 13:18:22] DEBUG p2p: Peer disconnected activePeers=13 multiAddr=/ip4/18.198.94.42/tcp/9100/p2p/16Uiu2HAm1D1E9vADzTLqU5Fbzms2XVwKmS8m3htFEFxeXoWVzMkn
[2020-12-16 13:18:22] DEBUG sync: Could not decode goodbye stream message error=stream reset peer=16Uiu2HAkysRukLUarWpd7WTs3tYSW94atWNuuvrWMomaXSV9Q5ar topic=/eth2/beacon_chain/req/goodbye/1/ssz_snappy
[2020-12-16 13:18:22] DEBUG sync: Could not decode goodbye stream message error=stream reset peer=16Uiu2HAmUakJSubLR8cePjboYqibQ4DzmKkwNXZg9SpvQLj2SWUo topic=/eth2/beacon_chain/req/goodbye/1/ssz_snappy
[2020-12-16 13:18:23] DEBUG sync: Could not decode goodbye stream message error=stream reset peer=16Uiu2HAm4oAhmrJvvAczSQoSxS2VtFjJdnU1vRyG2ywAggV3qaMT topic=/eth2/beacon_chain/req/goodbye/1/ssz_snappy
[2020-12-16 13:18:23] DEBUG p2p: Peer connected activePeers=14 direction=Outbound multiAddr=/ip4/100.0.28.3/tcp/9000/p2p/16Uiu2HAm9icCM9F1QFGe2pwoVRMSRQy4B6CRGQ6PQjnjo5VAWTim
[2020-12-16 13:18:23] DEBUG p2p: Peer disconnected activePeers=13 multiAddr=/ip4/100.0.28.3/tcp/9000/p2p/16Uiu2HAm9icCM9F1QFGe2pwoVRMSRQy4B6CRGQ6PQjnjo5VAWTim
[2020-12-16 13:18:24] DEBUG powchain: Latest eth1 chain event blockHash=0x076097d2bd2a2f2b22293712a40375433e8af2ebc1ccf09ad8b4815afae46d8e blockNumber=3936557
[2020-12-16 13:18:25] DEBUG sync: Could not decode goodbye stream message error=stream reset peer=16Uiu2HAm4dbQYsRUCF23z95sNHt2EGYrQaLHVr8bRAitnrWfAJ4S topic=/eth2/beacon_chain/req/goodbye/1/ssz_snappy
[2020-12-16 13:18:26] DEBUG sync: Could not decode goodbye stream message error=stream reset peer=16Uiu2HAmUaTfhABncpLndeMJvuJrXNbW4nG1QnJdx4Q7epC37eKY topic=/eth2/beacon_chain/req/goodbye/1/ssz_snappy
[2020-12-16 13:18:26] DEBUG sync: Could not decode goodbye stream message error=stream reset peer=16Uiu2HAmHMADkrbakQc6x7tMKo48sArUK67PGziW7qjRUAp5Qk46 topic=/eth2/beacon_chain/req/goodbye/1/ssz_snappy
[2020-12-16 13:18:29] DEBUG p2p: Peer connected activePeers=14 direction=Outbound multiAddr=/ip4/3.127.208.1/tcp/9100/p2p/16Uiu2HAmNJDtzswfRx1GTd8K1KuRcYicLJ4NnJFrKrjNguU3rffH
[2020-12-16 13:18:29] DEBUG p2p: Peer disconnected activePeers=13 multiAddr=/ip4/3.127.208.1/tcp/9100/p2p/16Uiu2HAmNJDtzswfRx1GTd8K1KuRcYicLJ4NnJFrKrjNguU3rffH
[2020-12-16 13:18:31]  INFO blockchain: Synced new block block=0xcb13522f... epoch=6302 finalizedEpoch=6300 finalizedRoot=0x43e50f61... slot=201692 slotInEpoch=28
[2020-12-16 13:18:31]  INFO blockchain: Finished applying state transition attestations=102 attesterSlashings=0 deposits=0 proposerSlashings=0 voluntaryExits=0
[2020-12-16 13:18:32] DEBUG sync: Could not decode goodbye stream message error=stream reset peer=16Uiu2HAm96LueUhG7uZ3tVYtZ536CAvLh6WRPcW1Q5QW2DuixPpv topic=/eth2/beacon_chain/req/goodbye/1/ssz_snappy
[2020-12-16 13:18:32] DEBUG p2p: Peer disconnected activePeers=12 multiAddr=/ip4/161.97.82.84/tcp/9000/p2p/16Uiu2HAm96LueUhG7uZ3tVYtZ536CAvLh6WRPcW1Q5QW2DuixPpv
[2020-12-16 13:18:32] DEBUG sync: Could not decode goodbye stream message error=stream reset peer=16Uiu2HAmT22tC5LUzifM2DkpRkqnC7A4dNoCxp9SScoRwb7v2fd topic=/eth2/beacon_chain/req/goodbye/1/ssz_snappy
[2020-12-16 13:18:32] DEBUG p2p: Peer disconnected activePeers=11 multiAddr=/ip4/45.14.151.133/tcp/9001/p2p/16Uiu2HAmT22tC5LUzifM2DkpRkqnC7A4dNoCxp9SScoRwb7v2fd
[2020-12-16 13:18:32] DEBUG sync: Could not decode stream message error=i/o deadline reached peer=16Uiu2HAmMvqKZLeKKZMrkjsqEp74rtr3yuuF1U1qbUbBYVv1DUCm topic=/eth2/beacon_chain/req/ping/1/ssz_snappy
[2020-12-16 13:18:33] DEBUG sync: Could not reset stream error=read data when expecting EOF topic=/eth2/beacon_chain/req/ping/1/ssz_snappy
[2020-12-16 13:18:34] DEBUG p2p: Peer connected activePeers=12 direction=Outbound multiAddr=/ip4/184.60.28.235/tcp/9001/p2p/16Uiu2HAmL9haThJNeGQaQrpHzx2aHSuimVQaH4mBnSZVoVisMxj5
[2020-12-16 13:18:38] DEBUG sync: Could not decode goodbye stream message error=stream reset peer=16Uiu2HAmBBvahQ6ofXt4p4DCbh59yveeuVhQue8Szx1aZTCe5Rvz topic=/eth2/beacon_chain/req/goodbye/1/ssz_snappy
[2020-12-16 13:18:38] DEBUG powchain: Latest eth1 chain event blockHash=0x39791b96a7d584d4510cf1c48ee4180ed5d4ac77ce33fdf1e21f82208a2923ce blockNumber=3936558
[2020-12-16 13:18:39] DEBUG p2p: Peer connected activePeers=14 direction=Outbound multiAddr=/ip4/162.230.216.51/tcp/9001/p2p/16Uiu2HAmJe6UQcZjZRvufFLRaSvjq9CSvftzhagHTQeSivjtvi4d
[2020-12-16 13:18:44]  INFO blockchain: Synced new block block=0x0a6fac86... epoch=6302 finalizedEpoch=6300 finalizedRoot=0x43e50f61... slot=201693 slotInEpoch=29
[2020-12-16 13:18:44]  INFO blockchain: Finished applying state transition attestations=113 attesterSlashings=0 deposits=0 proposerSlashings=0 voluntaryExits=0
```

The beacon-node will attempt to replace any reset connection with a new one. With a naive, single-threaded implementation of the responsive TCP-reset attack, the connectivity slowly degenerated but was however never brought down to 0. We will further investigate how the victim node's peer score attestation/proposal rates are affected.

Good job though, libp2p discovery!

However, the attack is easy to implement and scale. Furthermore, the attack is inexpensive enough to perform over an extended time on a series of block proposers and attesters. This makes it possible that no new blocks are getting gossipped, which threatens the liveness of the chain and can lead to inactivity penalties.

Notes & Nitpicks

Node can force to become Secio “preferred peer”

Location:
https://github.com/libp2p/specs/blob/master/secio/README.md#determining-roles-and-algorithms

Secio describes a proof-of-work-like scheme, where two hashes (“`oh1 < oh2`”) are compared to determine which cipher suites are available for an upcoming handshake. The hash is computed from a concatenation that includes a nonce, which may be arbitrarily chosen by an attacker. Therefore, an attacker can, with high likelihood, become the “`preferred peer`” in this scheme and force its selection of exchanges, suites, and hashes.

Prysm - http-web3 provider endpoint accepts http cleartext without warning

That’s not a sane default.

Eth2.0-deposit-CLI insecure default Keystore permissions generated by the

Location:

https://github.com/ethereum/eth2.0-deposit-cli/

The default permissions of generated validator keys are `644`.

- processes ran by other users on the machine can read the keystore
- processes ran by the same users can read/write the keystore (can be very bad)

Private key permissions should be set to `400`, similar to ssh keys.

Next Steps

Continuous beacon-fuzzing - Prysm

We conducted some 10^7 brute force TCP fuzzing iterations on prysm (fuzzotron + radamsa). A dedicated beacon fuzzing instance is currently being set up and planned to be available early 2020 for continuous fuzzing of the latest prysm.

Twist/Invalid Curve attacks on secp256k1

Several parties claimed secp256k1 is not twist secure. Implications on Geth/Libp2p should be understood and if found to be significant, the vulnerabilities should be further investigated.

Manual auditing - Prysm

For the upcoming two months, a large portion of time is set aside for manual auditing, in particular leveraging @protolamdas attacknets and rumor toolstack. Additionally, noise/secio protocol downgrades seem to be interesting leads

Social Engineering, Spear phishing - Ethereum.org

In particular, spear phishing is still one of the most prevalent threats to most organizations. Therefore, we recommend to conduct a targeted, Gsuite based (2fa-) credential harvesting spear-phishing campaign to collect data on how effective such an attack would be and further harden contributors' senses for this particular threat.