The Challenges of Blockchain-Based Naming Systems for Malware Defenders

Anonymous Authors

***Abstract*—Successful malware campaigns often rely on infected hosts' ability to locate and contact C2 servers. Malware campaigns often used DNS domains for this purpose, but DNS domains may be taken down by the registrar that sold them. In response to this threat, malware operators have begun using *blockchain-based naming systems* to store C2 server names. Blockchain naming systems are a threat to malware defenders because they are not subject to a centralized authority, such as a registrar, that can take down abused domains, either voluntarily or under legal pressure. In fact, blockchains are robust against a variety of interventions that work on DNS domains, which is bad news for defenders.**

We analyze the ecosystem of blockchain naming systems and identify new locations for defenders to stage interventions against malware. In particular, we find that malware is obligated to use centralized or semi-centralized infrastructure to connect to blockchain naming systems and modify the records stored within. In fact, scattered interventions have already been staged against this centralized infrastructure: we present case studies of several such instances. We also present a study of how blockchain naming systems are currently abused by malware operators, and discuss the factors that would cause a blockchain naming system to become an unstoppable threat. We conclude that existing blockchain naming systems still provide opportunities for defenders to prevent malware from contacting its C2 servers.

I. Introduction

Malware that is distributed across multiple hosts needs a way to distribute commands, upload stolen data, and coordinate between infected machines. Most malware, such as botnets or ransomware, uses a central command and control (C2) server for this task. However, as a single point of failure, a central C2 server presents an obvious weak link for defenders to target [1]. Malware authors must therefore be able to easily relocate and replace a C2 server after a defender takedown. Furthermore, all previously infected hosts must be able to find the new server at its new address, without outside coordination — if they cannot, they become useless. Malware authors avoid this "sunk cost" problem by providing a layer of indirection — a *naming layer* — instead of hard-coding a fixed address directly into deployed malware. This naming layer must be resilient to takedown efforts.

Until recently, the naming layer used most frequently by malware was ordinary DNS, which is rarely blocked at the protocol level, universally supported, and easy to configure. Malware authors use various strategies, such as DGAs (domain generation algorithms), to cycle through domains and complicate defense efforts. However, DNS domains are subject to centralized authorities such as registrars, who may be compelled to seize or deny access to abused domains. Malware authors have recently come up with an innovative solution to this risk: they have started to use *blockchain-based naming systems*.

Blockchain naming systems present several potential challenges for defenders. First, because they have no central authority to carry out legal takedown requests, they are immune to one of the most effective tools in malware defenders' arsenals. Second, some blockchain naming systems have high transaction costs to register and manage domains, which renders some existing defense strategies ineffective. For example, registering all the domains that a DGA can generate is impractical in expensive blockchain naming systems. However, blockchain naming systems present challenges to malware authors as well, such as the difficulty of stealthily accessing the blockchain.

We study five blockchain naming systems and the challenges and advantages that each present to malware authors and defenders. We argue that defender interventions are still possible for each of these systems, because name resolution requests must pass through centralized or partially centralized infrastructure to access any blockchain naming system. These centralized "chokepoints" still present viable locations for defenders to stage interventions. We also perform a measurement study of how malware is currently using these naming systems, and conclude that while some systems have seen significant abuse, others are currently much less likely to attract malware due to their high cost. We conclude that while blockchain naming systems present a significant threat, defenders still have viable options for enacting C2 takedowns.

II. Background

From a malware author's perspective, an ideal naming system for C2 addresses should have two properties: it should be difficult to censor specific individual names, and it should also be costly to take down or block the system as a whole. Typically, this first property, per-name takedown-resistance, is directly tied to the existence of a central authority with the ability to take down an individual record. System-wide takedown-resistance, on the other hand, is commonly a byproduct of a system's overall utility to legitimate users; potential collateral damage to benign users can create strong pressures against such takedowns.

To some extent, a trade-off exists between these features. For example, protocols such as Tor provide high resistance to the censorship of specific names, but because Tor traffic is both easily identified and represents a small volume of users, some defenders will simply block Tor altogether (i.e., under

the assumption that any damage to benign users will be minor). On the other side of the spectrum, malware has repurposed ubiquitous systems such as social media to store C2 addresses, because such traffic does not stand out and blanket bans on social media URLs are practically untenable. However, social media companies such as Facebook and Twitter have the capability, motivation, and (at times) legal obligation to remove abusive posts used to coordinate C2 activity. Thus, malware authors are incentivized to find naming systems that are neither vulnerable to censorship of individual records nor likely to be blocked or taken down entirely.

A. Tradeoffs of DNS-based C2 names

In part due to its ubiquity, DNS has been one of the most widely used naming systems for malware C2 servers. Because of its central role in the Internet's function, DNS naturally satisfies the system-wide takedown-resistance requirement; completely blocking DNS would be unthinkable for any modern enterprise or ISP. However, DNS is subject to a hierarchy of defined authorities, each of which may disable individual domain names, either voluntarily or in response to legal compulsion. For example, the registrar that sold a domain can delete it, or sinkhole it (i.e., divert its traffic) and prevent it from being subsequently updated or transferred. Similarly, the registry responsible for a domain name's top-level domain (TLD) can also take such actions.

Key to all these actions is that there is a singular legal entity with the capability to intervene. In some cases, they may do so voluntarily (e.g., such as when a registrar is notified of a violation of their terms of service.[1]) but they may also be compelled to take action by a court order. For example, US law enforcement may obtain a warrant to seize control of a domain name (subject to a showing of probable cause that it is involved in a criminal act), so long as the controlling registrar or registry is within US jurisdiction.[2] Typically such names are not then deleted (which would allow the malware author to re-register them), but instead the registrar (or registry) is compelled to redirect traffic to a benign site (aka a sinkhole) and block any attempts to change or transfer the domain by the registrant. Civil litigants can obtain similar effects via Temporary Restraining Orders (TRO) typically based on a showing that their intellectual property rights are being violated [1]. Microsoft, in particular, has made innovative use of trademark protection laws, such as the Lanham Act, to drive expansive private-sector botnet takedown efforts [3].

Today such legal takedowns are a critical tool for defenders to disable botnets. However, the effectiveness of this tool has led malware authors to respond by developing Domain Generation Algorithms (DGAs) which minimize the impact of any given takedown. When an infected host uses a DGA, it can randomly generate a large number of domains that its C2 server might be found at. DGAs usually use the current date as an input, which allows them to be kept in sync and changed on whatever time scale is convenient for the malware operators. Only a few of these domains will be registered at a time, to prevent defenders from preemptively taking them down. If a domain that is currently used to contact the C2 server is seized, the malware operators simply register a new one. Upon failing to reach the old C2 domain, every infected host will begin trying to resolve the rest of the names generated by the DGA until they discover the new, working C2 domain. Thus, to truly take down such a C2, defenders must register all of the domains (or otherwise prevent them from being registered) that the DGA can generate [4]. This intervention is costly in effort, but has been used successfully in the past [5].

To summarize, DNS-based C2 represent the status quo for modern botnets, but are vulnerable to concerted legal takedown efforts. What makes these takedowns possible is the existence of singular entities with the capability to unilaterally assert control over individual names *and* that those same entities are subject to legal jurisdictions available to defenders. Malware authors are thus incentivized to find naming systems that are not vulnerable to such efforts, either because the authority that can control the names is not in a takedown-amenable jurisdiction, or because no such authority exists.

B. Blockchain-based domain names

Blockchain-based naming systems present a potential threat because they claim to be immune to takedowns. This supposed immunity stems from several factors. First, no central authority controls blockchain domains in the same way that registrars control traditional DNS names. Unlike a DNS record, it is not generally possible to modify or delete a record on a blockchain without controlling the record's private key. Once a domain has been registered, its ownership is passed to the purchaser, after which point even the company that sold it cannot modify it.[3] Second, the machines running a blockchain are often distributed across so many countries and jurisdictions that seizing or taking down the entire system is prohibitively impractical. Third, once created, name records are stored immutably on the blockchain for as long as that blockchain exists, even if the owner later modifies or deletes them. With sufficient resources, and assuming all data is stored on-chain, it is possible to reconstruct the value of a blockchain-based naming record at any point in time, by parsing the transactions that modified the record's state up to that point.

These censorship-resistant properties are generally true under the assumption that an adversary has not found a way to compromise the entire blockchain, e.g. by gaining control of more than half of the blockchain's computational power. Such attacks are generally extremely difficult to execute. We focus instead on the common case where the blockchain underlying a naming system is not compromised, in which case the naming system as a whole is highly resistant to

[1]Registrars and hosting providers all typically specify a "Acceptable Use Policy" (AUP) in their contracts with third-parties. The details of these AUPs vary between providers, but it is common that they prohibit activity that is criminal, that violates intellectual property interests, or is highly disruptive.

[2]This process is described in considerable detail by Knight [2].

[3]This is true except in the case where the seller provides a domain parking service - we discuss this case in Section IV-F.

Name system	TLDs	Proxies
Namecoin	.bit	BDNS (defunct), PeerName
Emercoin	.lib, .bazar, .coin, .emc	friGate, PeerName, OpenNIC
Handshake	*any string*	hns.to, NextDNS, HDNS.io, BobWallet extension, LinkFrame extension
ENS	.eth	eth.link, eth.limo
Unstoppable	.crypto, .blockchain, .bitcoin, .coin, .nft, .wallet, .888, .dao, .x, .zil	Brave browser, Opera browser, Unstoppable browser, Unstoppable extension, Infura

TABLE I: Non-exhaustive selection of proxies, browsers, and extensions that can be used to access blockchain-based naming systems.

takedown efforts. Furthermore, some blockchains have become popular enough that even blocking access to them at a network level would cause collateral damage to licit users. Blockchains such as Bitcoin and Ethereum have recently skyrocketed in popularity as investors became interested in cryptocurrency as an asset class. As far as we are aware, cryptocurrencies and the blockchains they rely on are the first examples of strongly censorship-resistant systems that have gained a substantial community of legitimate users around the world.

Blockchain-based naming systems therefore provide both desirable properties of naming systems for C2 servers: it is difficult to take down the whole system as well as to take over individual records. Unfortunately, some malware is already aware of these advantages. BazarLoader uses Emercoin to record the domains of its C2 servers [6]. Namecoin is used by the Necurs botnet [7], the Chthonic banking trojan [8], Smoke Loader/Dofoil, Backdoor.Teamviewer, Shifu, and TinyNuke [9], [10]. Cerber ransomware has even used blockchain wallet addresses as names for its C2 servers [11].

III. Overview of Blockchain Naming Systems

In this section we present an overview of five blockchain-based naming systems, to provide background on how such systems work in detail. We select these systems based on their apparent popularity, as well as prior reports and literature that indicate some of them have already been abused by malware. These naming systems fall into two categories: systems built on naming-specific blockchains like Namecoin and Emercoin, whose purpose is primarily to store names and records, and systems built on general-purpose blockchains such as Ethereum, that are designed for purposes beyond naming systems. These systems also fall into two "generations:" Namecoin and Emercoin have existed since 2011 and 2013 respectively [12], [13], while the Ethereum Name Service (ENS), Handshake, and Unstoppable Domains are more recent inventions (2017, 2018, and 2019, respectively [14]–[16]).

All of the blockchain-based naming systems we study differentiate their names from DNS domains by creating alternate top level domains, which we refer to as *alt-TLDs* for brevity. A summary of the alt-TLDs used by each naming system is presented in Table I. Handshake names are slightly different, since the goal of the Handshake project is to replace the DNS root zone and make any alt-TLD available for purchase — see Section III-A2 for more details.

A. Naming-Specific Blockchains

We study three naming systems that are built on naming-specific and eponymous blockchains: Namecoin, Emercoin, and Handshake. All of these blockchains are primarily designed to support their naming systems, rather than to create new cryptocurrencies or support arbitrary blockchain-native programs ("smart contracts"). Because these blockchains have such specific purposes, they differ from blockchains like Ethereum in two ways: they have fewer participants and users, and their transaction fees are much less expensive. Both properties have implications for defenders — see Section IV for more details.

1) Namecoin and Emercoin: Namecoin and Emercoin, which are both modified copies of Bitcoin, are the oldest blockchain-based naming systems. Both were intended as additions to traditional DNS: users registered domains that resolved to IP addresses, using records very similar to DNS records. Unfortunately, Namecoin and Emercoin have been subject to a large amount of abuse. Four years after Namecoin's launch, Kalodner et al. found that only 28 of the 120,000 domains registered in Namecoin had meaningful web content, and most domain registrations appeared to be squatting [17]. In 2021, Casino et al. collected all of the IP addresses stored by Emercoin and Namecoin records, and submitted them to threat intelligence services including VirusTotal, Hybrid Analysis, Abuse.ch, and Pydnsbl (an aggregator of blocklists). They found that over 50% of the IPs in Namecoin and Emercoin records had been flagged as malicious by at least one threat intelligence service [18]. Furthermore, Casino et al. used a "poisoning" approach to find IP addresses associated with malicious IPs, either because the two addresses were stored in the same wallet, a name resolved to both addresses at different times, or the same email was recorded in their records. This "poisoning" approach revealed that the vast majority of IP addresses in Namecoin and Emercoin records are connected in some way to malicious IPs [18]. Our own findings support the conclusion that these naming systems are still rife with abuse (Section V).

2) Handshake: Handshake is a blockchain-based naming system that aims to replace the root DNS zone. It offers its users the ability to purchase nearly any string to use as an alt-TLD. Rather than selling second-level domains itself, Handshake allows its users to act as registrars who can sell

Record	Names with Record
Default NS and GLUE4 records	102,386
No A records	102,285
A 44.235.163.135	94
A 52.43.158.89	4
A 144.91.114.245	2
A 1.1.1.1	1
Invalid name	98,068
No record (null)	845
TXT record	138
"hello fx-wallet"	110
Other	28
Non-default NS record	32
Non-default GLUE4 record	11
Distributed storage address	7
Total unique names	201,458
Total records	201,487

TABLE II: Record types in the Handshake namespace.

their own domains. Handshake records are designed to store the NS records of traditional authoritative nameservers, rather than to replace DNS A, AAAA, or similar records. Handshake also allows users to store TXT records, which can contain the addresses for decentralized web hosting systems like Skynet [19] or IPFS [20]. Malware operators could potentially use Handshake as a naming system to find C2 content stored in these distributed storage systems. Additionally, Handshake advertises themselves as "the only naming blockchain with a lightweight recursive DNS resolver, which you can easily embed into browsers, apps, and devices" [21]. This lightweight resolver may be attractive to malware operators because it is small enough to be part of a malware payload.

To get a sense for how people use Handshake, we collected a sample of approximately 201,000 recently registered Handshake names by scraping a Handshake block explorer.[4] We attempted to scrape these names directly from the Handshake blockchain, but were unsuccessful because the RPC provided by the Handshake client to collect registered names from a Handshake node is no longer functional [22]. Table II summarizes our findings. At the moment, Handshake names appear to be overwhelmingly utilized as speculative assets. Only 0.14% of names in our sample had NS records that differed from the default. Of the names that kept the default nameserver and glue records, only 101 (0.05%) eventually resolved to A records, 94 of which were for the same IP address (a nameserver run by Namebase). Nearly half of registered Handshake domains in our sample cannot be resolved by the HNS client, since they contain illegal characters like emojis or are solely composed of numbers: these names are nevertheless allowed to be created on the Handshake blockchain. We concluded that the Handshake system has not yet seen significant adoption by either licit users or malicious actors.

B. Naming Systems on General Purpose Blockchains

Two naming systems based on the Ethereum blockchain have arrived since 2017: the Ethereum Name Service (ENS) and Unstoppable Domains. These naming systems are possible because of Ethereum's innovation in the blockchain space: *smart contracts.* Smart contracts are code that is embedded into the Ethereum blockchain. Any machine that runs an Ethereum "full node" can execute any smart contract. Each contract is identified by a 20-byte address, and makes its functions available through its Application Binary Interface (ABI). Thus, asking a smart contract to execute one of its functions is similar to making an RPC call, except that instead of one machine executing the code, every machine that receives the transaction must do so.

[4]https://e.hnsfans.com/names

Resolver Name	Txns Setting Resolver	Address
Public Resolver 2	33,304	`0x4976fb...`
Public Resolver 1	2,736	`0xDaaF96...`
OpenSea ENS resolver	482	`0x9C4e9C...`
ENS Old Public Resolver 2	440	`0x226159...`
Umbra: Stealth Resolver	409	`0xB37671...`
unnamed PublicResolver	126	`0xD3ddcC...`
unnamed PublicResolver	103	`0x5FfC01...`
ENS Old Public Resolver 1	29	`0x1da022...`

TABLE III: The ENS resolvers from which we collected a sample of names and records.

Smart contracts can be used to implement key-value stores, which means they are well suited to act as naming systems. For example, in a simplified system, a user might wish to set the name "foo.crypto" to resolve to the IP address 1.2.3.4. The user would create an Ethereum transaction that asks the key-value store's smart contract to call its "set record" function, with "foo.crypto" and "1.2.3.4" as function inputs. This transaction is then broadcast to the Ethereum network, and every Ethereum node that receives it updates its own copy of the key-value store to include the new record. Reading from the key-value store works similarly to writing to it: any Ethereum node can return a correct response. Notably, any transaction that causes a write costs a "gas fee" of Ethereum cryptocurrency. Gas fees are dependent on network congestion as well as other factors: they incentivize Ethereum node operators to execute smart contract code, which uses computing resources. In contrast, reading a smart contract's data does not cost a gas fee and does not create a transaction.

Interestingly, ENS and Unstoppable Domains are structured like DNS, but they are not necessarily being used as DNS replacements. The language and structure of both systems' smart contracts implies that they were modeled after DNS: for example, both systems use certain smart contracts as registries, and ENS even uses others as resolvers and registrars. However, users are primarily using these systems to map human-readable names to *cryptocurrency wallet addresses* instead of IP addresses. While users can still store IP addresses, traditional domains, TXT records, or distributed storage system (DS) addresses, very few choose to do so. This may imply that C2 records containing IP addresses or DNS domains will stand out and be easier for defenders to detect.

1) ENS: ENS names are registered (and resolved) in two steps involving two different smart contracts. First, a name

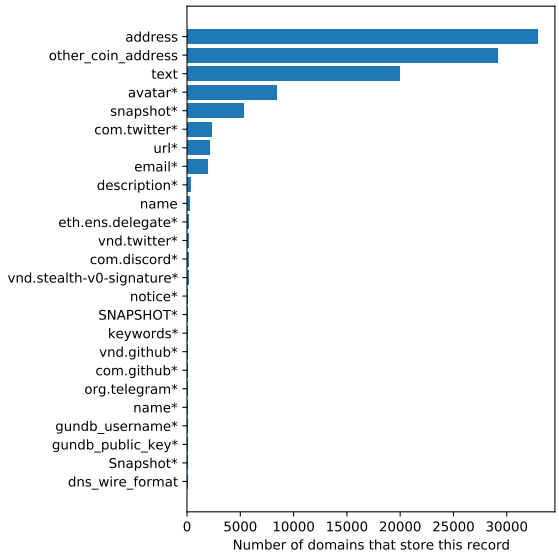

Fig. 1: Records stored by ENS names. *key within "text" record

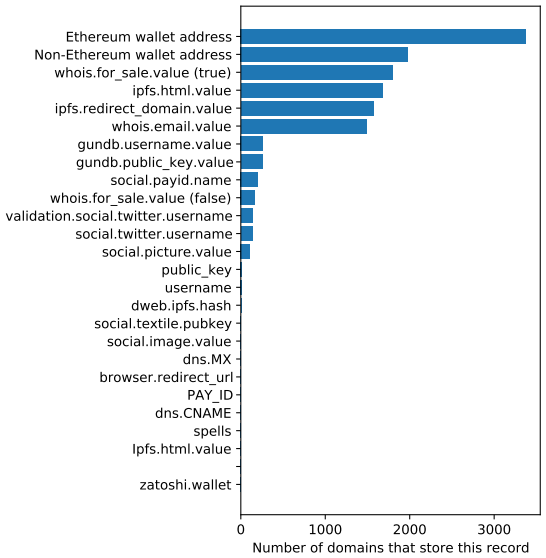

Fig. 2: Records stored by Unstoppable Domains names.

must be registered using the "ENS Registrar Controller" smart contract, which accepts the human-readable name and the address of a contract to use as a "resolver." Second, the resolver contract must be updated with the name's records. To complicate matters, names are not handled in their human-readable forms after they are registered: instead, they are referred to by their keccak256 hash. Furthermore, the ENS Registrar Controller contract allows users to specify a hash instead of a human-readable name, without ever performing a transaction that reveals the name itself. Therefore, to enumerate most of the names in ENS, we had to parse all of the transactions in the ENS Registrar Controller contract that recorded new hashes of names. We then queried the associated resolvers to discover the human-readable names. At the time of writing, at least 504 smart contracts had been set as resolvers for at least one name hash. We chose to take a sample of names from the eight resolver contracts that were set by the most names as their default resolver. The distribution of resolvers is long-tailed: the majority of resolvers resolve only a few names, while the eight most popular resolvers resolve the majority of names. We excluded addresses that were set as resolvers by many names but did not implement the ENS resolver specification, under the assumption that these were mistakes. Such misconfigured resolver addresses include the null address, 0x0, as well as unrelated smart contracts used by the ENS ecosystem. The resolvers we chose are detailed in Table III. This approach yielded a sample of 667,369 ENS names that were registered through the ENS Registrar Controller contract. Prior work has found that even after collecting all transactions from the ENS Registrar Controller and its historical predecessors, some hashes appear in the system but have never been seen to resolve to names [23]. It is unclear how these hashes came to exist, so we note that our sample does not contain all of the names in ENS, just the majority.

Figure 1 shows the distribution of the types of records stored in ENS for our sample of names. The majority resolve to wallet addresses or text records, not IP addresses, traditional domains, or DS addresses. We broke down the text records, which are key/value pairs, by the most common key names: these keys are marked with an asterisk. Only the most common 25 keys are shown. We note that only 17 names had `dns_wire_format` records, which are intended to store traditional DNS records, and all 17 are malformed as far as we can tell.

2) Unstoppable Domains: Like ENS, Unstoppable Domains uses Ethereum smart contracts as registrars. Unstoppable Domains names are divided into two systems. CNS (the Crypto Name System) contains all names with `.crypto` alt-TLDs, and has separate registry and resolver contracts. Later, Unstoppable added UNS (the Unstoppable Name System), which simplified name resolution by combining the resolver and registry contracts, and added several new alt-TLDs. Unstoppable Domains names never have to be renewed; they are purchased once and then owned indefinitely.

Like ENS names, Unstoppable Domains names are referenced by their hashes. We extracted all hashes from the UNS and CNS registry contracts by searching all of their transactions, and then found each hash's name and records by querying Unstoppable Domains' metadata endpoint.[5] This approach yielded a sample of 16,026 names. As with ENS, some names appear to exist in Unstoppable Domains that cannot be found by collecting transactions from these registry contracts. For example, it appears to be possible to store Unstoppable Domains names on the Polygon blockchain instead

[5]`https://metadata.unstoppabledomains.com/metadata/`

of Ethereum. We therefore note that our sample does not contain all of the names present in Unstoppable Domains.

Figure 2 shows the distribution of record types found in the Unstoppable Domains names. As in ENS, the majority of names have wallet records rather than records that point to websites in any way. The second most common type of record is "whois.for_sale.value," showing that many names are seen as speculative assets. Unstoppable Domains also provides an easy way for users to link to IPFS records.

We performed a web crawl of all of the Unstoppable names that had records pointing to websites, whether IPFS records, traditional IP addresses, or traditional domains. We took screenshots of the 367 websites we arrived at, inspected them manually, and did not find any evidence of malware use. Most websites were personal sites, Web3-based business sites, or related to the sale or collection of NFTs.

IV. Intervention Locations

Accessing any naming system, whether blockchain-based or DNS-based, requires a number of steps, each of which presents an opportunity for defenders to stage an intervention. Figure 3 compares the steps an infected host takes to resolve a DNS C2 domain (shown in orange) to the steps required to resolve a blockchain name (shown in green). While these steps involve different participants, parallels exist between the blockchain and DNS ecosystems. Figure 3 also details the interventions that can be staged by defenders at each step of the process. We now describe each step and its potential interventions in detail. For certain interventions at certain steps, we also present a case study of an attempt to stage the intervention in the wild, its results, and the lessons it provides for defenders.

A. Reaching the resolver

Regardless of whether a request is destined for DNS or a blockchain naming system, it must first reach the machine that acts as a resolver: either the DNS resolver or the proxy in Figure 3. Defenders may be able to intervene before this point by placing middleboxes with filter lists in the network. Some networks already have such defenses: for example, some ISP networks redirect all DNS requests to the ISP's own resolver, which can implement a filter list. This defense is probably not currently intended to block blockchain names, but it has that effect nevertheless in some cases. For example, some malware uses ordinary DNS rather than DNS-over-HTTPS to request blockchain domains, under the assumption that the proxy the query is intended for will redirect it to the blockchain naming system in the correct format. When ISPs perform DNS redirection to their own resolvers, these queries get redirected to the DNS root, which cannot resolve the alt-TLDs used by blockchain naming systems and return "NXDOMAIN." We present our study of this phenomenon in Section V. We observe that filter lists are only a partial defense against malware, because malware may utilize DGAs to evade them: as soon as a C2 name is added to the blocklist, the malware operators may register and begin using a new one.

B. Interventions at the name resolver

When resolving DNS domains, the entity that first attempts to respond to the request is a DNS resolver. When using blockchain naming systems, this entity is a proxy instead, shown in Figure 3 under "Name Resolvers." The proxy may expect queries in the form of DNS-over-HTTPS, unencrypted DNS, or in an arbitrary format. Instead of querying the DNS root zone, the TLD resolver, and eventually the authoritative nameserver to resolve a name, the proxy must connect to the blockchain and retrieve the record from one of the participants. Defenders may intervene at a traditional DNS resolver by requesting that the resolver implement a filter list, or the resolver operators may elect to implement one voluntarily. However, proxies that resolve blockchain names may be resistant to such voluntary efforts, because the blockchain ecosystem is often organized around principles of independence and self-governance, and resistance exists to the idea of censoring any content.

Proxies are currently the most common method for resolving blockchain names. Table I shows a selection of the proxies and tools that resolve names from each of the systems we study. The list of proxies is not exhaustive, but represents a subset of the best-known proxies in use at the time of writing. While most large browsers, such as Safari, Chrome, and Firefox, do not support any blockchain naming systems natively, some naming systems provide browser extensions that redirect blockchain name queries to proxies using DoH. A few browsers do resolve blockchain names without requiring extensions, such as Brave, which partners with a proxy called Infura [24]. Some naming systems have partnerships with existing DNS resolvers. For example, NextDNS's DNS resolvers can act as proxies to resolve Handshake names. Finally, some naming systems, such as Handshake, also provide stub resolver implementations that run locally on a user's computer. These stub resolvers also work by routing blockchain name queries to proxies.

Almost all of these proxies are centralized, in that they are controlled by a single authority. This is good news for defenders: similarly to traditional registrars, they are vulnerable to legal takedowns. They can be served with TROs or warrants and compelled to stop giving access to abused domains, as long as they operate within a jurisdiction amenable to such efforts. A centralized proxy could also be neutralized by serving a takedown order to its hosting provider, although this approach would produce varying amounts of the collateral damage depending on how many licit users utilize the proxy. While these interventions are not foolproof, they are subject to the same advantages and disadvantages as interventions on traditional registrars. Thus, centralized proxies return the distributed naming ecosystem to a state similar to the DNS ecosystem, from a defender's point of view.

1) Case Study #1 — OpenNIC ceasing support of `.bit`: OpenNIC is one of the few decentralized proxy services for blockchain names. It resolves names from several alternative naming systems, including Namecoin and Emercoin [32].

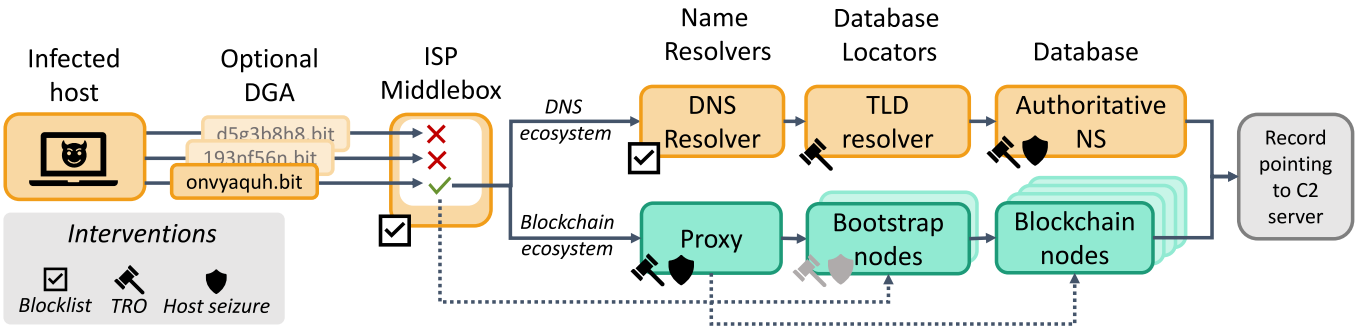

Fig. 3: Potential locations of interventions for blocking access to DNS-based and blockchain-based C2 server names.

OpenNIC's resolvers are run by a small community of volunteers [33]. In June 2019, this community voted of their own volition to remove support for Namecoin's `.bit` alt-TLD, because providers were beginning to block OpenNIC resolvers that were used by malware to resolve Namecoin names [34].

OpenNIC's decision to cease supporting Namecoin was not the result of a direct intervention by defenders, but it still yields an important lesson. Even a decentralized proxy service may be composed of few enough individuals that it is possible to cajole or compel them to stop resolving names used by malware. Furthermore, OpenNIC's community held this vote in response to pressure from Spamhaus, Malwarebytes, and other providers, who began blocklisting OpenNIC resolver domains: even if a proxy's operators cannot be contacted directly, it is still possible to pressure them to cease resolving names used by malware.

2) Case Study #2 — BDNS takedown: In April 2021, various defenders attempted to take down a proxy known as "Blockchain-DNS.info" or BDNS. BDNS reported on their website soon after the takedown attempt that seven of their domain names had been "un-delegated" and one of their API servers was shut down without warning [35]. For example, one endpoint, `bdns.io`, was apparently sinkholed by ShadowServer [36]. `bdns.io`'s NS records now point to variants of the name `sinkhole.shadowserver.org`.[6] We confirmed that these NS records were changed to ShadowServer's domains on March 26, 2021, using the pDNS database [37]. BDNS received a message from Spamhaus shortly after noticing the takedown, stating that several of BDNS's endpoint domains had been added to Spamhaus's blocklist. BDNS claimed that their browser extensions continued to resolve blockchain names using other endpoints, and directed users to a list of endpoints that were still working [38]. BDNS also stated that they had moved some infrastructure to a friendlier hosting provider, PRQ, which states on its website that "If [content] is legal in Sweden, we will host it, and will keep it up regardless of any pressure to take it down" [39]. However, as of August 2022, all of the endpoints listed in BDNS's Github repository [38] are either failing to resolve or resolving but failing to load content, and the proxy appears to be nonfunctional.

This takedown effort provides several lessons for defenders. First, defenders must take care when choosing a takedown strategy for a proxy. In this case, defenders tried two tactics: adding the proxy's endpoints to a widely used blocklist and taking down some domains and a hosting server entirely. The former tactic appeared to work well in locations where ISPs use Spamhaus's blocklist: BDNS stated that their proxy "may be still unreachable in those parts of the world." However, the domain takedown appeared to be only partially effective, since BDNS could still resolve blockchain names for a time using unaffected endpoints. We conclude that care must be taken to enumerate all of a proxy's endpoints and shut them down simultaneously.

C. Skipping the proxy: the rise of light clients

While proxies greatly simplify the process of connecting to a blockchain, they are not strictly necessary, which is bad news for defenders. We initially assumed that no infected host would be able to skip the proxy and participate directly in the blockchain, because acting as a blockchain node requires too many resources. However, this assumption turned out to be incorrect, because of the rise of *light clients*. When blockchains were first envisioned, most assumed that every participant in the network would be a "full" implementation of a node: it would contain enough state to reconstruct the entire history of the chain, all the way back to the first transaction. Additionally, each node would contribute to the blockchain by verifying every transaction it heard about. As blockchains grow over time, they become too resource-intensive to run on anything other than a dedicated, powerful machine. Two resources serve as the constraints: first, CPU power, which is obviously necessary to perform mining but now is even a bottleneck for transaction verification, because so many transactions happen per second. Second, disk space and speed: for example, a full Ethereum node cannot be run on a machine with a hard disk drive anymore, because nothing slower than a solid state drive can keep up with the reads and writes required [25]. These resource constraints make it very unlikely that malware could run "full" blockchain nodes on infected hosts. However, these constraints have also given

[6] sinkhole-0[0-4].shadowserver.org and sinkhole-[a-b].shadowserver.org.

rise to the concept of a "light client," a blockchain node with limited functionality that can fetch transactions from the chain but does not contribute by verifying transactions, mining, or broadcasting. Light clients are designed to run on laptops and mobile devices. As such, they use few enough resources to reasonably be included in malware.

D. Interventions at the Database Locator

Light clients enable malware to act as a first-class member of a blockchain, and discover other members of the chain using the chain's peer-to-peer discovery protocol without using a centralized proxy. In this case, defenders are left with a harder location to stage an intervention: the blockchain's bootstrap nodes, which is the blockchain equivalent of a service that locates the database of naming records. We show this path in Figure 3 with the dotted line between the middlebox and the bootstrap nodes.

In traditional DNS, the resolver must locate the database that contains a record by first querying the hierarchy of DNS servers: first the root and then the TLD resolver. The TLD resolver's role is to tell the DNS resolver which machine stores the database that ultimately contains a name's records. In a blockchain system, this role is filled by the bootstrap nodes. The purpose of the bootstrap nodes is to provide a gateway to the blockchain for new participants: new blockchain nodes find their initial list of potential peers by connecting to the bootstrap nodes. Blockchains use various methods to publish bootstrap node addresses for their users. For example, Ethereum uses a list of bootstrap nodes that are hard-coded into client implementations [26]. Bitcoin stores lists of bootstrap nodes in DNS TXT records maintained by volunteers, as well as hard-coded lists [27].

When defenders perform interventions by putting legal pressure on registrars, the intervention takes effect at the TLD resolver, which implements the changes to the zone file that affect the malware's domains. These changes can include "sinkholing" the domain by causing it resolve to an IP controlled by defenders or "freezing" it so that its records cannot be modified. This intervention does not translate well to blockchain naming systems for several reasons.

First, while bootstrap nodes are responsible for finding the entire naming database, they do not allow defenders to specify which blockchain systems a client may access and which it may not. This means that seizing a specific naming record, or even the entire naming system, is not possible at the bootstrap nodes. Consequently, disabling or seizing bootstrap nodes prevents all new clients from accessing any functionality provided by the blockchain, including the blockchain's cryptocurrencies and any services it offers unrelated to naming. This approach therefore carries the potential for a lot of collateral damage. Second, bootstrap nodes may be widely distributed across the globe, leading to jurisdictional challenges in bringing legal pressure to bear on their operators. Bootstrap nodes may also be difficult to find, since they may not be run by hosting providers but rather by anonymous individual volunteers. Third, bootstrap nodes may be numerous enough that finding and seizing them all may be prohibitively difficult. Finally, while the default bootstrap node lists are published for each blockchain, users may choose to substitute their own. A malware author could design a payload that contains an extensive list of machines that participate in a blockchain naming system, which would complicate a defender's efforts to take down all the potential participants. Because interventions at the bootstrap nodes are more challenging than interventions at the proxy, we show the intervention icons in gray in Figure 3.

However, defenders could fall back to using blocklists at network middleboxes to deny access to bootstrap nodes. For example, IDSes, enterprise firewalls, or ISP routers can drop traffic intended for bootstrap nodes. This approach is very similar to blocking any other malicious IP addresses, and is subject to the usual challenges. Defenders must keep blocklists up-to-date as malware authors update the IPs they connect to. To the advantage of defenders, any time malware authors are forced to update the IP addresses that bootstrap nodes may be found at, they run afoul of the "sunk cost" problem where infected machines that cannot be updated become useless. A similar argument applies if malware chooses to access bootstrap nodes using hard-coded DNS domain names instead of hard-coded IP addresses. Additionally, traditional interventions against domain names apply in that situation as well. Thus, while intervening at bootstrap nodes poses more of a challenge than intervening at centralized proxies, defenders still have viable options to choose from.

E. Interventions at the Database

In traditional DNS, defenders can sinkhole the domain of an authoritative nameserver or seize the server itself to prevent malware accessing a C2 domain record. This intervention is impractical for blockchain names, because instead of a single machine acting as the authoritative nameserver, every blockchain node has a copy of the database. Seizing the database would require either taking down every machine in the blockchain, or executing a successful "51%" attack by taking control of more than half of the computing power in the blockchain. Blockchains are generally highly robust against attacks like these, which makes them unlikely to be the most practical intervention for defenders to attempt. However, small naming-specific blockchains with few participants may be more vulnerable.

1) Case Study #3 — Namecoin's Vulnerability to 51% Attacks: Like all of the naming-specific blockchains that we study, the Namecoin blockchain has many fewer participants than blockchains like Ethereum. This makes Namecoin more vulnerable than larger blockchains to a "51%" attack. A 51% attack can be executed when an attacker controls more than half of the computational power of the blockchain, allowing them to rewrite historical transactions or add invalid ones. Gaining control of more than half of a blockchain's computational power is much easier on blockchains with few participants.

Namecoin has already experienced problems in this area. As of 2014, one mining pool known as "DiscusFish" or "F2Pool" consistently controlled more than 60% of the computational power of Namecoin, and on occasion controlled up to 75% [28]. While we did not find any reports that F2Pool had attacked Namecoin, they had the capability to do so. This vulnerability suggests two potential interventions. First, because Namecoin apparently has few licit users [18], interventions that render the entire naming system inoperable are more feasible than they would be on more popular general-purpose blockchains. Therefore, defenders could attempt to take over the entire blockchain and sinkhole all abused names by seizing control of F2Pool. Second, defenders could potentially apply legal pressure to the operators of F2Pool to coerce them to sinkhole certain specific names. This intervention would rely on defenders' ability to find F2Pool's operators and apply legal pressure in the jurisdiction the operators reside in. We predict that such an intervention would be challenging, but the fact that it appears possible at all contradicts the received wisdom that blockchains cannot be taken over directly.

F. Interventions after the name record is acquired

If an infected host successfully retrieves its C2 record, that record might take several forms. The three that we observed in existing blockchain naming systems that might be useful to malware were IP addresses, traditional DNS domains, and addresses for distributed storage systems like IPFS and SkyNet. Some naming systems also allow users to store arbitrary text as records, which would let malware operators store nonstandard record types like links to social media posts.

Each of these record types are subject to all of the traditional interventions that have already been described, except one: DS addresses. Distributed storage systems provide a form of "bulletproof" hosting, under the limitation that all hosted content must be static files and not dynamic websites. Any C2 server implemented entirely on such a system must be a simple file with no dynamic content. Infected hosts that wish to contact a distributed storage system must pass through the same steps shown in Figure 3 for accessing a blockchain, which means they are subject to the same interventions. For example, a strain of malware called "IPStorm" has already been discovered using IPFS for its C2 server in the wild. IPStorm connects to IPFS using bootstrap nodes [29], [30], which may be seized or blocked.

Another advantage for defenders is that some distributed storage systems, such as IPFS, do not have redundancy: only a single machine hosts each piece of a file. This raises the possibility of discovering the particular machine responsible for hosting a C2 server and seizing it.

A final possibility for intervening with the name record may be to seize names stored in "hosted" or "custodial" wallets. Some businesses, such as cryptocurrency exchanges, provide custodial wallets for users who wish to let the company handle their blockchain-based assets. This service is designed to make blockchain interaction easier for customers, but as a consequence, the business knows the custodial wallet's private key. If a name is stored in a custodial wallet, the business that runs the wallet could seize it [31]. However, a successful intervention must be difficult for malware operators to evade, and we note that malware operators with good operational practices can simply choose not to use custodial wallets.

G. Intervening with name modification or purchase

Generally speaking, DNS domains are cheaper, easier to modify, and easier to replace than IP addresses, because each IP address represents a compromised machine while new domains can be purchased inexpensively. Blockchain-based domains on general-purpose chains, such as Bitcoin and Ethereum, change this norm. While resolving a name is free, malware operators must pay *transaction fees* (known as *gas fees* on Ethereum) to register or modify names. These transaction fees can be quite expensive. For example, we found that registering a new name on the Unstoppable Domains service cost nearly $80 in gas fees during a period of high fees. In contrast, the cost of the name itself was $10. While licit users may wait for fees to be low at times of low network congestion, malware operators may not have that choice if they wish to avoid downtime in their campaign. High transaction costs poses challenges for defenders as well. For example, to combat DNS-based DGAs, defenders have the option of registering every domain the DGA will ever generate. This intervention would be much less practical if each registration was nearly an order of magnitude more expensive.

Naming-specific blockchains, such as Namecoin, Emercoin, and Handshake, present a different set of tradeoffs for defenders and malware operators. These blockchains are created with the sole intention of hosting a naming system. With fewer users and correspondingly less demand, these systems' names are usually much less expensive than names in Ethereum-based systems. This enables malware authors to use fast flux or DGA-based strategies, and also may enable defenders to pre-register domains generated by DGAs.

V. Measurements of Name Resolution Queries

Our analysis of the registered names in each blockchain naming system (Section III) indicated that malware is not yet utilizing ENS and Unstoppable Domains. In contrast, recent work on the records stored in Emercoin and Namecoin found that these systems were heavily used by malware as recently as 2020 [18]. However, to test whether malware is still using Namecoin and Emercoin and is not using ENS and Unstoppable Domains, it is necessary to analyze not only which names are *registered* in each system, but also which ones are heavily *used*. This is challenging because name resolutions are not transactions: they are read-only operations that do not leave a record on the blockchain. We cannot directly measure usage of blockchain names, but we observed that a side channel might exist to estimate name usage: "leakage" to the DNS. We predicted that since blockchain names require configuring alternate resolution systems, some requests might "leak" into the DNS when misconfigured machines attempt to resolve them as ordinary DNS domains. These leaked names would be

Malware	Domain	Lookups	Source
Gandcrab	malwarehunterteam.bit	348	[40]
	politiaromana.bit	341	[40]
	gdcb.bit	316	[40]
	zonealarm.bit	628	[41]
	ransomware.bit	1,039	[41]
CHESSYLITE	leomoon.bit	935	[42]
	lookstat.bit	710	[42]
	sysmonitor.bit	519	[42]
	volstat.bit	455	[42]
	xoonday.bit	573	[42]
Dofoil	vrubl.bit	988	[43]
	levashov.bit	1,059	[43]
	vinik.bit	6,265	[43]
KPOT Stealer	kpotuvorot10.bit	1,951	[44]
	star-fox.bit	351	[45]
Team9 Loader	bestgame.bazar	942	[46]
	forgame.bazar	865	[46]
	zirabuo.bazar	51	[46]
	tallcareful.bazar	146	[46]
	coastdeny.bazar	139	[46]
BazarLoader	acegikbcggin.bazar	546	[47]
	acegilbcggio.bazar	467	[47]
Trojan RTM	stat-counter-[0-9]-[0-9].bit	10,498	[48]
Necurs	jfbbrj3bbbd.bit	1,505	[49]
	qcmbartuop.bit	1,316	[7]

TABLE IV: Examples of malicious Namecoin and Emercoin domains in the October sample of B-root queries.

visible at the root DNS servers, but would not be forwarded to any other DNS servers, because the roots would respond that the alt-TLDs do not exist. An observer who could see which names were requested at a DNS root server with alt-TLDs corresponding to blockchain naming systems could get a sense for which names are in use.

We therefore took two samples of the names that were requested at the DNS B-root servers over the course of several days. The first sample consisted of names and how many requests were made for each on October 19, 2021. The second sample consisted of names, numbers of requests, and the ASes the requests were made from. It spanned two weeks in April 2022, from April 16 to April 30.

Another advantage of using B-root as a vantage point was that it let us observe requests for *unregistered* names that might indicate the presence of malware. DGAs work by generating a vast number of names, but only a few are ever registered and functional at any given time. These unregistered names do not, of course, appear in our samples of the registered names in each blockchain naming system. An infected host determines which names are registered by simply attempting to resolve them. If an infected host's queries were leaking into the DNS, we theorized that these queries would be very obvious, because the infected host would always receive NXDOMAIN responses from the root. These responses would cause the infected host to assume it has not found the correct C2 name for today and keep trying new names. The flood of nonexistent names with blockchain-based alt-TLDs would be visible when examining queries arriving at B-root.

A. Frequently Accessed Names

We first investigated how many days each name was requested on, and found that the vast majority of names are only requested once, on a single day. There were two notable exceptions: a small group (67) of `.bit` names that received a high volume of requests on every day of the sample, and a large group (39,000) of unique `.bazar` names that were each requested on most or all days of the sample. These names belong to the Namecoin and Emercoin naming systems, respectively.

We analyzed the group of `.bit` names that were requested on all 14 days of our sample and had more total requests than any name requested on fewer than 14 days. 66 names fit this criteria. We submitted them to VirusTotal and found that only 18 names were not labeled malicious by any engine, while 48 were labeled malicious by at least one.

The `.bazar` names that were requested on most (more than 10) days for our sample fell into two categories. The first contained names that appear to be generated by concatenating four lowercase-letter bigrams consisting of a consonant and a vowel (e.g., `acbaelek.bazar`, `acbaelel.bazar`, `acbaelid.bazar`). These names appear to be generated by the malware BazarLoader [50]. The second category contains 38 names that do not appear to be randomly generated. We uploaded these to VirusTotal and determined that 23 were labeled as malicious by at least one threat intelligence service, three were not indexed by VirusTotal, and the remainder were not labeled as malicious. Six of the names were themed around Australian tourism, of which four were labeled malicious and two were not: these names may also be associated with BazarLoader [51].

We note that the most popular Emercoin and Namecoin names each day were largely known to be associated with malware, which we determined by manually searching the Internet. We present a sample of the most popular malware-related names in Table IV. These names were taken from the October sample; the days in April had a similarly high number of malicious names that received high volumes of requests.

B. Unregistered ENS and Unstoppable Domains names

We observed a large number of names, mostly randomly generated names, with alt-TLDs that are used by ENS and Unstoppable Domains. However, we found that these names are actually unrelated to blockchain naming systems and are likely not part of malware campaigns. We drew this conclusion for two reasons. First, the randomly generated names only had one lookup each, and all of these lookups originated from a single AS (AS15169, Google). This is in contrast to lookups for randomly generated names in Emercoin and Namecoin that are known to be part of malware campaigns: these requests originate from many different ASes and some names receive many more than just one request. Second, not a single randomly generated domain with an ENS or Unstoppable Domains alt-TLD was registered in a blockchain naming system. If these names had been part of a malware campaign, at least one should have resolved to the address

of a C2 server at some point. It is possible that B-root only received failed requests from a single misconfigured machine, but this does not match the behavior we observe for malware campaigns that abuse Namecoin and Emercoin.

We predict that rather than being intended for use in a blockchain naming system, these randomly generated names were leaked from local networks, and were never intended to be resolved by either a blockchain naming system *or* the DNS root. A prior study on root DNS queries found that some networks use non-ICANN TLDs internally, under the assumption that queries for those names will never reach external DNS resolvers. However, these queries frequently leak to external networks [52]. We predict that some internal networks use alt-TLDs that coincidentally overlap with the blockchain naming systems' alt-TLDs. We concluded that these names were unlikely to be part of DGA-based malware campaigns, and were also likely unrelated to blockchain naming systems at all.

C. Requests for registered names from ENS and Unstoppable Domains

Very few registered names from ENS or Unstoppable Domains leaked to B-root: we observed fewer than 400 unique ENS names per day and fewer than 300 unique daily names from Unstoppable Domains. These names also received few requests per day compared to the names from Namecoin and Emercoin. No name received more than approximately 350 lookups, in contrast with the most popular domains in Namecoin, which received an order of magnitude more requests per day. We submitted every ENS and Unstoppable Domains name that received more than ten daily requests to VirusTotal. None were in VirusTotal's database, in contrast with names from Emercoin and Namecoin, which were largely present and flagged as malicious.

Each of these findings regarding names that leak to B-root support our conclusion that malware still heavily utilizes older systems like Namecoin and Emercoin, but has not yet adopted new systems like ENS or Unstoppable Domains. We predict that this is due to two factors. First, the monetary cost of creating and modifying names in ENS and Unstoppable Domains is much higher than in naming-specific systems like Namecoin and Emercoin. Second, defenders have apparently not yet been able to exert enough pressure on Namecoin and Emercoin to make these systems unattractive to malware operators, because we still see high malware usage of those systems. We hope that the findings in this work will aid defenders in exerting more effective pressure against malware operators.

VI. Discussion

Out of the five naming systems we examine, we find none so far that present an entirely intractable problem for defenders. For a naming system to present a threat, it must be both easily usable by malware authors and popular enough that blocking its bootstrap nodes, or blocking access to it entirely, will cause significant collateral damage to licit users. For a system to be widely adopted by licit users, it must have three necessary characteristics.

First, the system's name management must be as easy or easier than name management on traditional DNS domains. Users must not be required to write code themselves to interact with smart contracts, as is currently the case with each of the systems we study if the user does not use a custodial wallet. Users also must not be required to run a blockchain node in order to manage their names, as Handshake currently requires to the best of our knowledge.

Second, the transactions that are required to register and update names must be affordable. Transactions on Ethereum, in our experience, cost anywhere between $60 and $140 during the course of our experiments, although we discovered that we were attempting to make transactions during periods of high network congestion and fees were unusually high. Even transaction fees as low as ten dollars per transaction are far less affordable than transaction fees on naming-specific chains, which can be as low as a few cents. This dynamic may make ordinary users more likely to embrace naming systems built on naming-specific chains, rather than general-purpose chains. However, general-purpose chains may be better known, and therefore more likely to be trusted by users even if transaction fees are higher than on naming-specific chains. A trade-off may therefore exist between affordability and perceived trustworthiness and name recognition.

Third, licit users are unlikely to embrace any naming system that does not have widespread browser adoption. Browser adoption is hindered by naming systems' lack of coordination, which currently leads to name collisions: for example, the alt-TLDs `.wallet`, `.coin`, and `.x` are currently used by multiple blockchain naming systems. Some newly created ICANN TLDs also collide with Handshake TLDs, such as `.music`. Naming collisions present a barrier to browser adoption because the browser would either have to enforce some sort of precedence for systems that include colliding names, or users would have to choose which naming system to use for each name with collisions. Either option will confuse and frustrate users who are unfamiliar with the concepts of namespaces. So far, only browsers that focus on privacy as one of their primary features have chosen to resolve alternate naming systems, and none have chosen to resolve systems that might collide with either each other or ICANN TLDs. Until browsers can resolve an alternate naming system natively, users are unlikely to adopt that naming system.

We conclude that the higher ease of use of purchasing, modifying, and resolving traditional DNS domains is a very high barrier for blockchain-based naming systems to overcome. As long as blockchain naming systems are not widely adopted, we predict they will not become entirely intractable problems for defenders.

VII. Related Work

Kalodner et al. performed the first study to our knowledge of Namecoin in 2015 [17]. They conclude that the Namecoin ecosystem was "dysfunctional:" only 28 out of 120,000

registered names were valid, not squatted, and had nontrivial content.

Patsakis et al. present an analysis of potential weaknesses and user risks of Namecoin and Emercoin, including the risks of squatting, 51% attacks, phishing, and abuse by malware [53]. The authors also provide an overview of the names stored in these systems, and found that many names registered in the Alexa Top 1K were also registered under Namecoin and Emercoin's alt-TLDs. Most of these squatted names redirected to pornographic websites.

Casino et al. analyzed the IP addresses in Namecoin and Emercoin records [18]. They first identified malicious IP addresses using several threat intelligence databases, and then clustered all the IPs into "malicious," "suspicious" and "benign" categories with a "poisoning" approach. An IP was labeled "malicious" if a threat intelligence database categorized it as such. It was labeled "suspicious" if it appeared in the same wallet, was resolved to by the same domain, or shared the same email TXT record as a malicious IP, and "benign" if it had no connection to a malicious IP. Casino et al. discovered that only 8% of the IPs in Emercoin and 28% of those in Namecoin had no association with malicious IP addresses. While this paper mentioned the existence of more recent blockchain naming systems, it did not perform an analysis of any system except Namecoin and Emercoin.

Numerous other blockchain-based naming systems have been proposed, including the Blockstack Naming System [28], Bitforest/Conifer [54], [55], BlockDNS [56], and Nebulis [57]. To our knowledge, only Blockstack has evolved into a commercial product. We excluded the Blockstack Naming System from this work because it does not appear to be as popular as the other systems we study.

Other work has analyzed the ways in which blockchain technologies in general might be abused by malware. Pletinckx et al. analyzed Cerber ransomware and found that it used blockchain wallet addresses as domains [58]. Hassan et al. point out that blockchain nodes reside in so many different legal jurisdictions, it will be difficult for regulators to control what information gets passed across country borders [59]. Moubarak et al. present a theoretical design for malware to store pieces of its payload on Bitcoin [60].

Relatively little work has been done on defenses against malware that uses blockchain naming systems. Huang et al. developed a machine learning-based detection method for distinguishing malicious blockchain-based names from benign names in DNS traffic [61]. Hu et al. presented a brief comparison of DNS and Bitcoin-based naming systems, and noted that small, naming-specific blockchains like Namecoin were vulnerable to 51% attacks [62].

Prior work has evaluated the effectiveness of interventions that target DNS domains. Kesari et al. provide an overview of legal intervention methods and cites their use in a number of malware takedowns [1]. Wang et al. studied the use of TROs to seize storefronts run by spammers [63]. Liu et al. analyzed the effectiveness of two interventions that were initiated by registrars and designed to stop spammers from registering domains [64]. Prior literature has also analyzed interventions based on taking down hosting providers, and concluded that these interventions have modest or mixed effectiveness [65]–[68].

VIII. Conclusion

While decentralized naming and hosting systems pose challenges, they cannot entirely eliminate their reliance on systems with centralized authority. Whenever malware uses a centralized resource to enable its use of decentralized ones, defenders can intervene. Defenders cannot serve legal takedown orders to a centralized registrar to take down a blockchain domain, but they can prevent malware from accessing the blockchain in the first place, or target the DNS domain or IP address that the blockchain domain resolves to. We examined existing blockchain-based naming systems and found that naming systems on general purpose blockchains are not currenly attractive to malware because of their high cost. In contrast, systems on naming-specific blockchains present an ongoing threat, but these systems are susceptible to defenses such as blocklisting every IP address stored in the name records, blocking the proxies that resolve the names, or blocking the system entirely, because so little licit content exists on those blockchains. We conclude that for a naming system to be truly more dangerous than DNS, it must achieve widespread adoption as well as inexpensive transactions and high ease-of-use, and no existing naming systems have yet achieved all three characteristics.

References

[1] A. Kesari, C. Hoofnagle, and D. McCoy, "Deterring Cybercrime: Focus on Intermediaries," *Berkeley Technology Law Journal*, vol. 32, no. 3, pp. 1093–1134, 2017. [Online]. Available: https://heinonline.org/HOL/P?h=hein.journals/berktech32&i=1137

[2] J. Knight, "Domain Takedowns A Step by Step Analysis for Law Enforcement," Feb. 2015.

[3] Z. Lerner, "MICROSOFT THE BOTNET HUNTER: THE ROLE OF PUBLIC-PRIVATE PARTNERSHIPS IN MITIGATING BOTNETS," *Harvard Journal of Law and Technology*, vol. 28, no. 1, p. 26, 2014.

[4] M. Antonakakis, R. Perdisci, Y. Nadji, N. Vasiloglou, S. Abu-Nimeh, W. Lee, and D. Dagon, "From throw-away traffic to bots: Detecting the rise of dga-based malware," *USENIX Security*, p. 16, 2012.

[5] D. Piscitello, "ConfickerSummaryandReview20100507," p. 18.

[6] A. Brandt, "BazarLoader deploys a pair of novel spam vectors," Apr. 2021. [Online]. Available: https://news.sophos.com/en-us/2021/04/15/bazarloader/

[7] "The DGAs of Necurs." [Online]. Available: https://bin.re/blog/the-dgas-of-necurs/

[8] "Malware-Traffic-Analysis.net - 2016-12-27 - EITest Rig-E from 185.156.173.99 sends Chthonic banking Trojan," December 2016. [Online]. Available: https://www.malware-traffic-analysis.net/2016/12/27/index.html

[9] "abuse.ch | .bit - The next Generation of Bulletproof Hosting." [Online]. Available: https://abuse.ch/blog/dot-bit-the-next-generation-of-bulletproof-hosting/

[10] M. Mackie, "CryptoDNS–Should We Worry?" [Online]. Available: https://insights.sei.cmu.edu/blog/cryptodns-should-we-worry/

[11] S. Pletinckx, C. Trap, and C. Doerr, "Malware Coordination using the Blockchain: An Analysis of the Cerber Ransomware," in *2018 IEEE Conference on Communications and Network Security (CNS)*, May 2018, pp. 1–9.

[12] "Namecoin." [Online]. Available: https://www.namecoin.org/

[13] "Official website of Emercoin." [Online]. Available: https://emercoin.com/en/

[14] "Ethereum Name Service." [Online]. Available: https://ens.domains

[15] "Handshake." [Online]. Available: https://handshake.org

[16] etherscan.io, "Ethereum Transaction Hash (Txhash) Details | Etherscan." [Online]. Available: http://etherscan.io/tx/0x66d1300ef612ec633ede2f26fca4a17d5321ee4c0642d111b5092ec2efae90ba

[17] H. Kalodner, M. Carlsten, P. Ellenbogen, J. Bonneau, and A. Narayanan, "An empirical study of Namecoin and lessons for decentralized namespace design."

[18] F. Casino, N. Lykousas, V. Katos, and C. Patsakis, "Unearthing malicious campaigns and actors from the blockchain DNS ecosystem," *Computer Communications*, pp. 217–230.

[19] "Decentralized Internet for a Free Future | Skynet Labs." [Online]. Available: https://skynetlabs.com/

[20] "IPFS Powers the Distributed Web." [Online]. Available: https://ipfs.io

[21] "Access Handshake names." [Online]. Available: https://learn.namebase.io/starting-from-zero/how-to-access-handshake-sites

[22] "Node RPC getnames Response Exceeds Limit · Issue #447 · handshake-org/hsd." [Online]. Available: https://github.com/handshake-org/hsd/issues/447

[23] P. Xia, H. Wang, Z. Yu, X. Liu, X. Luo, G. Xu, and G. Tyson, "Challenges in decentralized name management: The case of ens," in *Proceedings of the 2013 Internet Measurement Conference (IMC)*, 2022.

[24] "Resolve Methods for Unstoppable Domains brave/brave-browser Wiki." [Online]. Available: https://github.com/brave/brave-browser

[25] "FAQ | Go Ethereum." [Online]. Available: https://geth.ethereum.org/docs/faq#wait-so-i-cant-use-fast-sync-on-an-hdd

[26] "ethereum/go-ethereum," Sep. 2022, original-date: 2013-12-26T13:05:46Z. [Online]. Available: https://github.com/ethereum/go-ethereum/blob/511bf8f18801520bf4e0c7e4d098a17d0665bc89/params/bootnodes.go

[27] "P2P Network — Bitcoin." [Online]. Available: https://developer.bitcoin.org/devguide/p2p_network.html

[28] M. Ali, J. Nelson, R. Shea, and M. J. Freedman, "Blockstack: A global naming and storage system secured by blockchains," in *2016 USENIX annual technical conference (USENIX ATC 16)*, 2016, pp. 181–194.

[29] "The InterPlanetary Storm: New Malware in Wild Using InterPlanetary File System's (IPFS) p2p network." [Online]. Available: https://www.anomali.com/blog/the-interplanetary-storm-new-malware-in-wild-using-interplanetary-file-systems-ipfs-p2p-network

[30] "This unusual Windows malware is controlled via a P2P network." [Online]. Available: https://www.zdnet.com/article/this-unusual-windows-malware-is-controlled-via-a-p2p-network/

[31] R. Pegoraro, "The blockchain is making domain names more private—for good or bad," Oct. 2021. [Online]. Available: https://www.fastcompany.com/90686579/blockchain-domains-bit-microsoft

[32] "OpenNIC Project." [Online]. Available: https://www.opennic.org/

[33] "OpenNIC Public Servers." [Online]. Available: https://servers.opennicproject.org/

[34] "Should OpenNIC drop support for NameCoin [OpenNIC Wiki]." [Online]. Available: https://wiki.opennic.org/votings/drop_namecoin

[35] "Blockchain-DNS.info – Blockchain Name Resolver," Jun. 2021. [Online]. Available: https://web.archive.org/web/20210621223622/https://blockchain-dns.info/#laghaus

[36] "The Shadowserver Foundation." [Online]. Available: https://www.shadowserver.org/

[37] DomainTools, "Iris Investigation Platform - Passive DNS," DomainTools, Jan. 2022, https://www.domaintools.com/products/iris.

[38] "GitHub - B-DNS/Resolver: Resolver implementation," Feb. 2021. [Online]. Available: https://web.archive.org/web/20210228044252/https://github.com/B-DNS/Resolver/

[39] "PRQ - Colocation, Dedicated Servers, Web hosting, VPN Tunnels, Privacy services." Jun. 2021. [Online]. Available: https://web.archive.org/web/20210623012111/https://prq.se/

[40] Trellix, "Trellix Insights: GandCrab ransomware version 2 released with new .crab extension and other changes," August 2022. [Online]. Available: https://kcm.trellix.com/corporate/index?page=content&id=KB93103&locale=en_US

[41] R. Tiwari, "Evolution of GandCrab Ransomware," August 2018. [Online]. Available: https://www.acronis.com/en-us/blog/posts/gandcrab/

[42] R. EITZMAN, K. GOODY, and J. VALDEZ, "How the Rise of Cryptocurrencies Is Shaping the Cyber Crime Landscape: Blockchain Infrastructure Use." [Online]. Available: https://www.mandiant.com/resources/blog/how-rise-cryptocurrencies-shaping-cyber-crime-landscape-blockchain-infrastructure-use

[43] M. D. S. R. Team, "Hunting down Dofoil with Windows Defender ATP," Apr. 2018. [Online]. Available: https://www.microsoft.com/security/blog/2018/04/04/hunting-down-dofoil-with-windows-defender-atp/

[44] S. Kim, "Deep analysis of KPOT Stealer," Jul. 2021. [Online]. Available: https://medium.com/s2wblog/deep-analysis-of-kpot-stealer-fb1d2be9c5dd

[45] J. S. LLC, "Automated malware analysis report for bz.arm-20211111-0250 - generated by joe sandbox," November 2021. [Online]. Available: https://www.joesandbox.com/analysis/519713/0/pdf

[46] N. Pantazopoulos and S. Antenucci, "In-depth analysis of the new Team9 malware family," Jun. 2020. [Online]. Available: https://blog.fox-it.com/2020/06/02/in-depth-analysis-of-the-new-team9-malware-family/

[47] "A Baza Valentine's Day | Proofpoint US," Feb. 2021. [Online]. Available: https://www.proofpoint.com/us/blog/threat-insight/baza-valentines-day

[48] "Own research, what can open source tell us? / Sudo Null IT News," Jun. 2021. [Online]. Available: https://web.archive.org/web/20210625041125/https://sudonull.com/post/3301-Own-research-what-can-open-source-tell-us

[49] Threatcrowd.org, "Malware: Win32.trojan-dropper.necurs." [Online]. Available: https://www.threatcrowd.org/listMalware.php?antivirus=Win32.Trojan-dropper.Necurs.Dzts

[50] J. Bader, "Yet Another Bazar Loader DGA," January 2021. [Online]. Available: https://bin.re/blog/yet-another-bazarloader-dga/

[51] "AlienVault - Open Threat Exchange." [Online]. Available: https://otx.alienvault.com/pulse/60fd3f0d396edd67255e401f

[52] Q. A. Chen, E. Osterweil, M. Thomas, and Z. M. Mao, "MitM Attack by Name Collision: Cause Analysis and Vulnerability Assessment in the New gTLD Era," in *2016 IEEE Symposium on Security and Privacy (SP)*, May 2016, pp. 675–690, iSSN: 2375-1207.

[53] C. Patsakis, F. Casino, N. Lykousas, and V. Katos, "Unravelling Ariadne's Thread: Exploring the Threats of Decentralised DNS," *IEEE Access*, vol. 8, pp. 118 559–118 571, 2020, conference Name: IEEE Access.

[54] Y. Dong, W. Kim, and R. Boutaba, "Bitforest: a portable and efficient blockchain-based naming system," in *2018 14th International Conference on Network and Service Management (CNSM)*. IEEE, 2018, pp. 226–232.

[55] ——, "Conifer: centrally-managed pki with blockchain-rooted trust," in *2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*. IEEE, 2018, pp. 1092–1099.

[56] S. Ren, B. Liu, F. Yang, X. Wei, X. Yang, and C. Wang, "Blockdns: Enhancing domain name ownership and data authenticity with blockchain," in *2019 IEEE Global Communications Conference (GLOBECOM)*, 2019, pp. 1–6.

[57] R. Kastelein, "Blockchain Startup Nebulis Set to Solve Problem of DDoS Attacks On DNS Servers," Dec. 2016. [Online]. Available: https://www.the-blockchain.com/2016/12/06/blockchain-startup-nebulis-set-prevent-ddos-attacks-dns-servers/

[58] S. Pletinckx, C. Trap, and C. Doerr, "Malware coordination using the blockchain: An analysis of the cerber ransomware," in *2018 IEEE Conference on Communications and Network Security (CNS)*, 2018, pp. 1–9.

[59] F. u. Hassan, A. Ali, M. Rahouti, S. Latif, S. Kanhere, J. Singh, AlaAl-Fuqaha, U. Janjua, A. N. Mian, J. Qadir, and J. Crowcroft, "Blockchain And The Future of the Internet: A Comprehensive Review," Nov. 2020, arXiv:1904.00733 [cs]. [Online]. Available: http://arxiv.org/abs/1904.00733

[60] J. Moubarak, M. Chamoun, and E. Filiol, "Developing a k-ary malware using blockchain," in *NOMS 2018 - 2018 IEEE/IFIP Network Operations and Management Symposium*, Apr. 2018, pp. 1–4, iSSN: 2374-9709.

[61] Z. Huang, J. Huang, and T. Zang, "Leopard: Understanding the threat of blockchain domain name based malware," in *International Conference on Passive and Active Network Measurement*. Springer, 2020, pp. 55–70.

[62] H. Wei-hong, A. Meng, S. Lin, X. Jia-gui, and L. Yang, "Review of blockchain-based dns alternatives," *Chinese Journal of Network and Information Security*, vol. 3, no. 3, pp. 71–77.

[63] D. Y. Wang, M. Der, M. Karami, L. Saul, D. McCoy, S. Savage, and G. M. Voelker, "Search + Seizure: The Effectiveness of Interventions on SEO Campaigns," in *Proceedings of the 2014 Conference on*

Internet Measurement Conference - IMC '14. Vancouver, BC, Canada: ACM Press, 2014, pp. 359–372. [Online]. Available: http://dl.acm.org/citation.cfm?doid=2663716.2663738

[64] H. L. Liu, K. Levchenko, M. Félegyházi, C. Kreibich, G. Maier, G. M. Voelker, and S. Savage, "On the Effects of Registrar-level Intervention," p. 8.

[65] D. Bradbury, "Testing the defences of bulletproof hosting companies," *Network Security*, vol. 2014, no. 6, pp. 8–12, 2014.

[66] M. Konte, R. Perdisci, and N. Feamster, "Aswatch: An as reputation system to expose bulletproof hosting ases," in *Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication*, 2015, pp. 625–638.

[67] A. Noroozian, J. Koenders, E. v. Veldhuizen, C. H. Ganan, S. Alrwais, D. McCoy, and M. v. Eeten, "Platforms in Everything: Analyzing Ground-Truth Data on the Anatomy and Economics of Bullet-Proof Hosting," 2019, pp. 1341–1356. [Online]. Available: https://www.usenix.org/conference/usenixsecurity19/presentation/noroozian

[68] S. Alrwais, X. Liao, X. Mi, P. Wang, X. Wang, F. Qian, R. Beyah, and D. McCoy, "Under the shadow of sunshine: Understanding and detecting bulletproof hosting on legitimate service provider networks," in *2017 IEEE Symposium on Security and Privacy (SP)*, 2017, pp. 805–823.