The Sybil Attack

John R. Douceur
Microsoft Research
johndo@microsoft.com

"One can have, some claim, as many electronic personas as one has time and energy to create."
– *Judith S. Donath* [12]

Abstract – *Large-scale peer-to-peer systems face security threats from faulty or hostile remote computing elements. To resist these threats, many such systems employ redundancy. However, if a single faulty entity can present multiple identities, it can control a substantial fraction of the system, thereby undermining this redundancy. One approach to preventing these "Sybil attacks" is to have a trusted agency certify identities. This paper shows that, without a logically centralized authority, Sybil attacks are always possible except under extreme and unrealistic assumptions of resource parity and coordination among entities.*

1. Introduction

We[*] argue that it is practically impossible, in a distributed computing environment, for initially unknown remote computing elements to present convincingly distinct identities. With no logically central, trusted authority to vouch for a one-to-one correspondence between entity and identity, it is always possible for an unfamiliar entity to present more than one identity, except under conditions that are not practically realizable for large-scale distributed systems.

Peer-to-peer systems commonly rely on the existence of multiple, independent remote entities to mitigate the threat of hostile peers. Many systems [3, 4, 8, 10, 17, 18, 29, 34, 36] *replicate* computational or storage tasks among several remote sites to protect against integrity violations (data loss). Others [5, 6, 7, 16, 28] *fragment* tasks among several remote sites to protect against privacy violations (data leakage). In either case, exploiting the redundancy in the system requires the ability to determine whether two ostensibly different remote entities are actually different.

If the local entity has no direct physical knowledge of remote entities, it perceives them only as informational abstractions that we call *identities*. The system must ensure that distinct identities refer to distinct entities; otherwise, when the local entity selects a subset of identities to redundantly perform a remote operation, it can be duped into selecting a single remote entity multiple times, thereby defeating the redundancy. We term the forging of multiple identities a *Sybil attack* [30] on the system.

It is tempting to envision a system in which established identities vouch for other identities, so that an entity can accept new identities by trusting the collective assurance of multiple (presumably independent) signatories, analogous to the PGP web of trust [37] for human entities. However, our results show that, in the absence of a trusted identification authority (or unrealistic assumptions about the resources available to an attacker), a Sybil attack can severely compromise the initial generation of identities, thereby undermining the chain of vouchers.

Identification authorities can take various forms, not merely that of an explicit certification agency such as VeriSign [33]. For example, the CFS cooperative storage system [8] identifies each node (in part) by a hash of its IP address. The SFS network file system [23] names remote paths by appending a host identifier to a DNS name. The EMBASSY [22] platform binds machines to cryptographic keys embedded in device hardware. These approaches may thwart Sybil attacks, but they implicitly rely on the authority of a trusted agency (such as ICANN [19] or Wave Systems [35]) to establish identity.

In the following section, we define a model of a distributed computing environment that lacks a central authority. Building on this model, Section 3 proves a series of lemmas that severely limit the ability of an entity to determine identity. Section 4 surveys related work, and Section 5 concludes.

[*] Use of the plural pronoun is customary even in solely authored research papers; however, given the subject of the present paper, its use herein is particularly ironic.

2. Formal model

As a backdrop for our results, we construct a formal model of a generic distributed computing environment. Our model definition implicitly limits the obstructive power of corrupt entities, thereby strengthening our negative results. The universe, shown schematically in Fig. 1, includes:

- A set *E* of infrastructural *entities e*
- A broadcast communication *cloud*
- A *pipe* connecting each entity to the cloud

Set *E* is partitioned into two disjoint subsets, *C* and *F*. Each entity *c* in subset *C* is *correct*, abiding by the rules of any protocol we define. Each entity *f* in subset *F* is *faulty*, capable of performing any arbitrary behavior except as limited by explicit resource constraints. (The terms "correct" and "faulty" are standard in the domain of Byzantine fault tolerance [21], even though terms such as "honest" and "deceptive" might be more appropriate.)

Entities communicate by means of *messages*. A message is an uninterrupted, finite-length bit string whose meaning is determined either by an explicit protocol or by an implicit agreement among a set of entities. An entity can send a message through its pipe, thereby broadcasting it to all other entities. The message will be received by all entities within a bounded interval of time. Message delivery is guaranteed, but there is no assurance that all entities will hear messages in the same order.

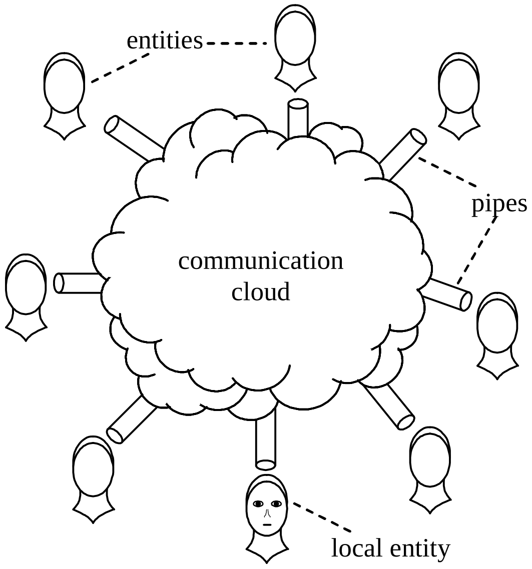

Fig. 1: Formal model of distributed environment

This model has two noteworthy qualities: First, it is quite general. By leaving the internals of the cloud unspecified, this model includes virtually any interconnection topology of shared segments, dedicated links, routers, switches, or other components. Second, the environment in this model is very friendly. In particular, in the absence of resource constraints, denial-of-service attacks are not possible. A message from a correctly functioning entity is guaranteed to reach all other correctly functioning entities.

We place a minimal restriction on the relative computational resources available to each entity, namely that there exists some security parameter *n* for which all entities can perform operations whose computational complexity is (low-order) polynomial in *n* but for which no entity can perform operations that are superpolynomial in *n*. This restriction allows entities to use public-key cryptography [24] to establish virtual point-to-point communication paths that are private and authenticated. Although these virtual paths are as secure as point-to-point physical links, they come to exist only when created by pairs of entities that have acknowledged each other. Our model excludes direct links between entities because a physical link provides a form of centrally supplied identification of a distinct remote entity. Also, in the real world, packets can be sniffed and spoofed, so the base assumption of a broadcast medium (augmented by cryptography) is not unrealistic.

An *identity* is an abstract representation that persists across multiple communication events. Each entity *e* attempts to *present* an identity *i* to other entities in the system. (Without loss of generality, we state our results with respect to a specific local entity *l* that is assumed to be correct.) If *e* successfully presents identity *i* to *l*, we say that *l accepts* identity *i*.

A straightforward form for an identity is a secure hash of a public key. Under standard cryptographic assumptions, such an identifier is unforgeable. Furthermore, since it can generate a symmetric key for a communication session, it is also persistent in a useful way.

Each correct entity *c* will attempt to present one *legitimate* identity. Each faulty entity *f* may attempt to present a legitimate identity and one or more *counterfeit* identities. Ideally, the system should accept all legitimate identities but no counterfeit entities.

3. Results

This section presents four simple lemmas, with nearly trivial proofs, that collectively show the impracticality of establishing distinct identities in a large-scale distributed system.

An entity has three potential sources of information about other entities: a trusted agency, itself, or other (untrusted) entities. In the absence of a trusted authority, either an entity accepts only identities that it has directly validated (by some means) or it also accepts identities vouched for by other identities it has already accepted.

For direct validation, we show:

- Even when severely resource constrained, a faulty entity can counterfeit a constant number of multiple identities.
- Each correct entity must simultaneously validate all the identities it is presented; otherwise, a faulty entity can counterfeit an unbounded number of identities.

Large-scale distributed systems are inevitably heterogeneous, leading to resource disparities that exacerbate the former result. The latter result presents a direct impediment to scalability.

For indirect validation, in which an entity accepts identities that are vouched for by already accepted identities, we show:

- A sufficiently large set of faulty entities can counterfeit an unbounded number of identities.
- All entities in the system must perform their identity validations concurrently; otherwise, a faulty entity can counterfeit a constant number of multiple identities.

Since the number of faulty entities in the system is likely to grow as the system size increases, the former result places another limit on system scale. The latter restriction becomes harder to satisfy as system size increases.

3.1. *Direct identity validation*

The only direct means by which two entities can convince a third entity that they are distinct is by performing some task that a single entity could not. If we assume that the resources of any two entities differ by at most a constant factor, a local entity can demand proof of a remote entity's resources before accepting its identity. However, this leaves us with the following limitation:

Lemma 1: If ρ is the ratio of the resources of a faulty entity *f* to the resources of a minimally capable entity, then *f* can present $g = \lfloor \rho \rfloor$ distinct identities to local entity *l*.

Proof: Define r_M as the resources available to a minimally capable entity. By hypothesis, *g* entities can present *g* identities to *l*; therefore, g r_M resources are sufficient to present *g* identities. Since $\rho \geq g$, *f* has at least $g\ r_M$ resources available, so it can present *g* identities to *l*.

Lemma 1 states a lower bound on the damage achievable by a faulty entity. To show how this can be enforced as an upper bound, we present three mechanisms that can (at least theoretically) exploit limitations in three different resources: communication, storage, and computation.

If communication resources are restricted, local entity *l* can broadcast a request for identities and then only accept replies that occur within a given time interval.

If storage resources are restricted, entity *l* can challenge each identity to store a large amount of unique, uncompressible data. By keeping small excerpts of this data, entity *l* can verify, with arbitrarily high probability, that all identities simultaneously store the data they were sent.

If computation resources are restricted, entity *l* can challenge each identity to solve a unique computational puzzle. For example, the local entity can generate a large random value *y* and challenge the identity to find, within a limited time, a pair of values *x*, *z* such that the concatenation *x* | *y* | *z*, when run through a secure hash function, yields a value whose least significant *n* bits are all zero:

$$\text{given } y\text{, find } x, z \text{ s.t. } \mathrm{LSB}_n(\mathrm{hash}(x \mid y \mid z)) \;=\; 0$$

The time to solve[*] such a puzzle is proportional to 2^{n-1}. The time to verify the result is constant. (The reason for allowing the challenged entity to find a prefix *x* and a suffix *z*, rather than merely one or the other, will become clear in Section 3.2.)

[*] measured in count of hash function evaluations. For a random oracle [2] hash function, the only way to find a solution is to iterate through candidate values of *x* and/or *z*; compute the hash for each *x* | *y* | *z* triple; and test the result. Actual implementation requires a hash function that is both preimage-resistant and resistant to non-brute-force attacks such as chaining attacks [24].

To be effective, these resource challenges must be issued to all identities simultaneously:

Lemma 2: If local entity *l* accepts entities that are not validated simultaneously, then a single faulty entity *f* can present an arbitrarily large number of distinct identities to entity *l*.

Proof: Faulty entity *f* presents an arbitrarily long succession of distinct identities to *l*. The resources required for each presentation are used and then freed for the subsequent presentation.

Lemma 2 is insurmountable for intrinsically temporal resources, such as computation speed and communication bandwidth. However, since storage is not inherently time-based, entity *l* can indefinitely extend the challenge duration by periodically demanding to see newly specified excerpts of the stored data. If an accepted identity ever fails to meet a new challenge, the local entity can discard it from its acceptance list, thereby eventually catching a Sybil attack that it might have initially missed. A major practical problem with this extension is that (by assumption) the challenge consumes the majority of an entity's storage resources, so extending the challenge duration greatly impedes the ability of the entity to perform other work. (However, the challenge data itself could be valuable data, compressed and encrypted by the local entity before sending it to the remote entities, using a different key for each remote entity to maintain challenge uniqueness.)

3.2. *Indirect identity validation*

As described in the introduction, the reason for establishing the distinctness of identities is to allow the local entity to employ redundancy in operations it delegates to remote entities. One such operation it could conceivably delegate is the validation of other identities. Thus, in addition to accepting identities that it has directly validated using one of the challenge mechanisms described above, an entity might also accept identities that have been validated by a sufficient count of other identities that it has already accepted.

If an entity that has presented identity i_1 claims to have accepted another entity's identity i_2, we say that i_1 *vouches for* i_2. An obvious danger of accepting indirectly validated identities is that a group of faulty entities can vouch for counterfeit identities:

Lemma 3: If local entity *l* accepts any identity vouched for by *q* accepted identities, then a set *F* of faulty entities can present an arbitrarily large number of distinct identities to *l* if either $|F| \ge q$ or the collective resources available to *F* at least equal those of $q + |F|$ minimally capable entities.

Proof: Define r_F as the total resources available to set *F*, r_k as the resources available to each faulty entity f_k, and r_M as the resources available to a minimally capable entity. Then:

$$q + |F| \le \frac{r_F}{r_M} = \sum_{f_k \in F} \frac{r_k}{r_M} < \sum_{f_k \in F} \left\lfloor \frac{r_k}{r_M} \right\rfloor + |F|$$

By Lemma 1, entity f_k can present $\lfloor r_k / r_M \rfloor \ge 1$ identities to *l*, so *F* can present *q* identities to *l*. Thereafter, all of *F*'s identities vouch for an arbitrarily large number of counterfeit identities, all of which will be accepted by *l*.

As in the case of direct identity validation, indirect identity validation also has a concurrency requirement. In particular, all entities must perform their resource challenges concurrently:

Lemma 4: If the correct entities in set *C* do not coordinate time intervals during which they accept identities, and if local entity *l* accepts any identity vouched for by *q* accepted identities, then even a minimally capable faulty entity *f* can present $g = \lfloor |C| / q \rfloor$ distinct identities to *l*.

Proof: Define r_M as the resources required to present one identity. By assumption, entity *f* has r_M resources available. Partition set *C* into *g* disjoint subsets C_k of minimum cardinality *q*. Faulty entity *f* presents identity i_k to each entity in C_k, using r_M resources during time interval T_k. Since T_k need not overlap with $T_{k'}$ for $k \ne k'$, r_M resources are available during interval $T_{k'}$ to present identity $i_{k'} \ne i_k$ to entities in set $C_{k'}$. At least *q* entities in each set C_k will vouch for distinct identity i_k, so *l* will accept all *g* identities.

Lemma 4 shows the need for multiple entities to issue challenges concurrently. Whether it is possible for a correct entity to satisfy multiple concurrent challenges depends upon the resource:

In our formal model, all communication is broadcast, so an entity can simultaneously reply to communication challenges from arbitrarily many entities. (However, this is rather less practical in actual networks than in our abstract model.)

It may not be possible to satisfy multiple concurrent storage challenges, and there are information-theoretic reasons for believing that it is impossible, since every bit of data stored for one challenger consumes one bit of storage space that is thus unavailable to serve another challenger (and the data from all challengers is, of necessity, incompressible). This may prevent storage challenges from being used for indirect validation.

For computation challenges, it is possible for an entity to solve multiple puzzles simultaneously by combining them. If an entity receives m puzzles $y_1, y_2, \ldots y_m$, it can find a w such that:

$$\mathrm{LSB}_n(\mathrm{hash}(0 \mid y_1 \mid y_2 \mid \ldots y_m \mid w)) = 0$$

Then, the solution to each puzzle y_k is:

$$x_k = 0 \mid y_1 \mid y_2 \mid \ldots y_{k-1} \text{ and } z_k = y_{k+1} \mid \ldots y_m \mid w$$

An obvious danger here is that if a validating entity issues challenges to multiple identities that have been counterfeited by a single faulty entity, the faulty entity could combine the challenges and solve them together. However, the challenger can identify this attempted Sybil attack by checking whether $x_1 \mid y_1 \mid z_1 = x_2 \mid y_2 \mid z_2$ for any two solutions from putatively different identities.

Like Lemma 1, the result of Lemma 4 is that a faulty entity can amplify its influence. A system that can tolerate a fraction φ of all identities being faulty can tolerate only φ/g of all entities being faulty. In some systems, this may be acceptable.

4. Related work

Most prior research on electronic identities has focused on persistence and unforgeability [14, 15, 27, 31], rather than on distinctness.

Computational puzzles are an old technique [25] that has become popular recently for resisting denial-of-service attacks [1, 9, 20] by forcing the attacker to perform more work than the victim.

Dingledine et al. [11] suggest using puzzles to provide a degree of accountability in peer-to-peer systems, but this still allows a resourceful attacker to launch a substantial attack, especially if the potential for damage is disproportionate to the fraction of the system that is compromised.

The issue of establishing on-line identities for humans has been studied for some time [12, 32], with solutions that generally depend on some direct interaction in the physical world [13, 37].

5. Summary and conclusions

Peer-to-peer systems often rely on redundancy to diminish their dependence on potentially hostile peers. If distinct identities for remote entities are not established either by an explicit certification authority (as in Farsite [3]) or by an implicit one (as in CFS [8]), these systems are susceptible to Sybil attacks, in which a small number of entities counterfeit multiple identities so as to compromise a disproportionate share of the system.

Systems that rely upon implicit certification should be acutely mindful of this reliance, since apparently unrelated changes to the relied-upon mechanism can undermine the security of the system. For example, the proposed IPv6 privacy extensions [26] obviate much of the central allocation of IP addresses assumed by CFS.

In the absence of an identification authority, a local entity's ability to discriminate among distinct remote entities depends on the assumption that an attacker's resources are limited. Entities can thus issue resource-demanding challenges to validate identities, and entities can collectively pool the identities they have separately validated. This approach entails the following conditions:

- All entities operate under nearly identical resource constraints.
- All presented identities are validated simultaneously by all entities, coordinated across the system.
- When accepting identities that are not directly validated, the required number of vouchers exceeds the number of system-wide failures.

We claim that in a large-scale distributed system, these conditions are neither justifiable as assumptions nor practically realizable as system requirements.

Acknowledgements

The author thanks Miguel Castro for issuing the challenge that led to this paper, Jon Howell and Sandro Forin for reviewing drafts, Dan Simon for sanity checking the combinable computational puzzle described in Section 3.2, the anonymous IPTPS reviewers for their helpful suggestions for improving this presentation, and Brian Zill for suggesting the term "Sybil attack."

References

[1] T. Aura, P. Nikander, J. Leiwo, “DoS-Resistant Authentication with Client Puzzles”, *Cambridge Security Protocols Workshop*, Springer, 2000.

[2] M. Bellare and P. Rogaway, “Random Oracles are Practical: A Paradigm for Designing Efficient Protocols”, *1st Conference on Computer and Communications Security*, ACM, 1993, pp. 62-73.

[3] W. J. Bolosky, J. R. Douceur, D. Ely, M. Theimer, “Feasibility of a Serverless Distributed File System Deployed on an Existing Set of Desktop PCs”, *SIGMETRICS 2000*, 2000, pp. 34-43.

[4] M. Castro, B. Liskov, “Practical Byzantine Fault Tolerance”, *3rd OSDI*, 1999.

[5] D. Chaum, “Untraceable Electronic Mail, Return Addresses, and Digital Pseudonyms”, *CACM* 4 (2), 1982.

[6] B. Chor, O. Goldreich, E. Kushilevitz, M. Sudan, “Private Information Retrieval”, *36th FOCS*, 1995.

[7] I. Clarke, O. Sandberg, B. Wiley, T. Hong, “Freenet: A Distributed Anonymous Information Storage and Retrieval System”, *Design Issues in Anonymity and Unobervability*, ICSI, 2000.

[8] F. Dabek, M. F. Kaashoek, D. Karger, R. Morris, I. Stoica, “Wide-Area Cooperative Storage with CFS”, *18th SOSP*, 2001, pp. 202-215.

[9] D. Dean, A. Stubblefield, “Using Client Puzzles to Protect TLS”, *10th USENIX Security Symp*., 2001.

[10] R. Dingledine, M. Freedman, D. Molnar “The Free Haven Project: Distributed Anonymous Storage Service”, *Design Issues in Anonymity and Unobservability*, 2000.

[11] R. Dingledine, M. J. Freedman, D. Molnar “Accountability”, *Peer-to-Peer: Harnessing the Power of Disruptive Technologies*, O’Reilly, 2001.

[12] J. S. Donath, “Identity and Deception in the Virtual Community”, *Communities in Cyberspace*, Routledge, 1998.

[13] C. Ellison, “Establishing Identity Without Certification Authorities”, *6th USENIX Security Symposium*, 1996, pp. 67-76.

[14] U. Feige, A. Fiat, A. Shamir, “Zero-Knowledge Proofs of Identity”, *Journal of Cryptology* 1 (2), 1988, pp. 77-94.

[15] A. Fiat, A. Shamir, “How to Prove Yourself: Practical Solutions of Identification and Signature Problems”, *Crypto ’86*, 1987, pp. 186-194.

[16] Y. Gertner, S. Goldwasser, T. Malkin, “A Random Server Model for Private Information Retrieval”, *RANDOM ’98*, 1998.

[17] A. Goldberg, P. Yianilos, “Towards an Archival Intermemory”, *International Forum on Research and Technology Advances in Digital Libraries*, IEEE, 1998, pp. 147-156.

[18] J. H. Hartman, I. Murdock, T. Spalink, “The Swarm Scalable Storage System”, *19th ICDCS*, 1999, pp. 74-81.

[19] ICANN, Internet Corporation for Assigned Names and Numbers, 4676 Admiralty Way, Suite 330, Marina del Rey, CA 90292-6601, www.icann.org.

[20] A. Juels, J. Brainard, “Client Puzzles: A Cryptographic Defense against Connection Depletion Attacks”, *NDSS ’99*, ISOC, 1999, pp. 151-165.

[21] L. Lamport, R. Shostak, M. Pease, “The Byzantine Generals Problem”, *TPLS* 4(3), 1982.

[22] K. R. Lefebvre, “The Added Value of EMBASSY in the Digital World”, Wave Systems Corp. white paper, www.wave.com, 2000.

[23] D. Mazières, M. Kaminsky, M. F. Kaashoek, E. Witchel, “Separating Key Management from File System Security”, *17th SOSP*, 1999, pp. 124-139.

[24] A. J. Menezes, P. C. van Oorschot, S. A. Vanstone. *Handbook of Applied Cryptography*. CRC Press, 1997.

[25] R. C. Merkle, “Secure Communications over Insecure Channels”, *CACM 21*, 1978, pp. 294-299.

[26] T. Narten, R. Draves, “Privacy Extensions for Stateless Address Autoconfiguration in IPv6”, *RFC 3041*, 2001.

[27] K. Ohta, T. Okamoto, “A Modification to the Fiat-Shamir Scheme”, *Crypto ’88*, 1990, pp. 232-243.

[28] M. K. Reiter, A. D. Rubin, “Crowds: Anonymous Web Transactions”, *Transactions on Information System Security* 1 (1), ACM, 1998.

[29] A. Rowstron, P. Druschel, “Storage Management and Caching in PAST, a Large-Scale, Persistent Peer-to-Peer Storage Utility”, *18th SOSP*, 2001, pp. 188-201.

[30] F. R. Schreiber, *Sybil*, Warner Books, 1973.

[31] A. Shamir, “An Efficient Identification Scheme Based on Permuted Kernels”, *Crypto ’89*, 1990, pp. 606-609.

[32] S. Turkle, *Life on the Screen: Identity in the Age of the Internet*, Simon & Schuster, 1995.

[33] VeriSign, Inc. 487 East Middlefield Road, Mountain View, CA 94043, www.verisign.com.

[34] M. Waldman, A. D. Rubin, L. F. Cranor, “Publius: A Robust, Tamper-Evident Censorship-Resistant Web Publishing System”, *9th USENIX Security Symposium*, 2000, pp. 59-72.

[35] Wave Systems Corp. 480 Pleasant Street, Lee, MA 01238, www.wave.com

[36] J. J. Wylie, M. W. Bigrigg, J. D. Strunk, G. R. Ganger, H. Kilite, P. K. Khosla, “Survivable Information Storage Systems”, *IEEE Computer* 33 (8), IEEE, 2000, pp. 61-68.

[37] P. Zimmerman, *PGP User’s Guide*, MIT, 1994.